Adaptive Image Filtering for DCT Artifact Reduction

A THESIS

Presented to

The Academic Faculty

By

Irving Linares

In Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering

Georgia Institute of Technology

May 1998

Adaptive Image Filtering
for DCT Artifact Reduction

Approved:

____
Russell M. Mersereau, Co-Chairman

____
Mark J. T. Smith, Co-Chairman

____
James H. McClellan

Date approved by Co-Chairmen ______________

Dedicated to my beloved wife Dorah,
and to our wonderful children
Verónica, Juan Carlos, and Hugo.

Acknowledgments

There are many people to whom I want to express my gratitude for their help and support during the course of this research. In particular, I am most grateful to my advisors Dr. Russell M. Mersereau, and Dr. Mark J. T. Smith. Their encouragement, patience, and guidance over the years has made this contribution possible. It has been a real pleasure and honor to be one of their privileged students and to learn from their vast scholarly experience. I am very grateful to Dr. James McClellan for his chairmanship of my dissertation proposal and for his participation on my thesis defense committee. I am thankful to Dr. Erik Verriest for being a member of the proposal and the defense committees as well. I am very grateful to Dr. Christopher Heil of the School of Mathematics for serving on my thesis defense committee. I am also thankful to Dr. G. Tong Zhou for her participation on my dissertation proposal committee.

At Georgia Tech I also had the opportunity to learn from world-class professors such as Ronald W. Schafer, Vijay Madisetti, and Douglas B. Williams. I am deeply indebted to my advisors, to Dr. Dale C. Ray, Vice Chair of The School of Electrical and Computer Engineering, and to the NASA Goddard Space Flight Center for their continued support under my very special circumstances as an in-absentia part-time graduate student at Georgia Tech. Also, I want to thank the Center for Signal and Image Processing secretaries, Kay Gilstrap and Stacy Schultz for every little thing they made to help me present the ideas proposed in this thesis. I am very grateful to our support staff, in particular, Dave Webb, Lonnie D. Harvel, and Steve Flynn.

Their expediency in troubleshooting any network problems that prevented remote access to the Electrical and Computer Engineering network has been invaluable.

Many fellow students were very helpful and generous by providing some of the algorithms used in this research and by helping me directly with their comments and suggestions. Mathew Crouse, now at the University of Texas in Austin and previously at the University of Illinois in Urbana-Champaign provided the *adaptQ()* algorithm for JPEG image-adaptive quantization table computation. Jose G. González at the Georgia Institute of Technology, kindly provided the Set Partitioning In Hierarchical Trees (SPHIT) subband coding algorithms and the necessary guidance to modify them for our purposes. Also Muhammed Coban and Pham Quoc provided data, programs and insight into several of the algorithms presented in this research.

I also must thank several former graduate students of the Center for Signal and Image Processing who have also significantly contributed with their suggestions, comments, software, and encouragement. I am especially grateful to Steve Martucci, Wilson Chung, Faouzi Kossentini, Alen Docef, Halûk Aydinoğlu, Ragnar Jónsson, and Dinei Florencio. I would like to thank the Center for Signal and Image Processing at Georgia Tech for providing the resources and computing environment that made possible this research.

At the NASA Goddard Space Flight Center, Joe Stevens and Dr. Mareboyana Manohar were always available to discuss many of the ideas presented in this thesis. I am very grateful for their invaluable feedback and for their help with a number of subjective quality tests. I also want to thank the GSFC Part-time Graduate Study Program, the GSFC Research and Study Fellowship Program, the Office of Human Resources, the GSFC Homer E. Newell Memorial Library, the former Mission Operations and Data Systems Directorate and the former Networks Division, now the Network and Mission Services Project, for their continued support during these years. Without GSFC's support and computing facilities, this work would have not been possible.

Many fellow, former, and NASA-related people also need to be mentioned. In one way or another they kindly contributed to this research. In no particular order, I want to express my sincere gratitude to Keiji Tasaki, Diane Scheuerman, Phil Ryan, Wayne Boswell, Paul Ondrus, Dan Spintman, Dr. Don Hei, Les Wentz, Bob Spearing, Dr. Adolfo Figueroa-Viñas, Dr. Jim Tilton, Phil Liebrecht, Bill Watson, Dick Nafzger, Tom Sardella, Madeline Butler, Gilberto Colón, Rosa Acevedo, Pat Hennessy, Oscar González, Larry Jackson, John Donohue, Badri Younes, Pat Schwabenbauer, Dave Zillig, Frank Stocklin, and Betsy Edwards. Thanks also to our very close personal friends José Casals and Lorraine Morell. To all those coworkers, and to my very special friends from the University of Puerto Rico, please accept the most sincere thanks for your encouragement and support.

I also want to thank all my teachers and professors throughout my academic formation. Some of them, in a very intense way, have instilled in me the pursuit of excellence in all my endeavors.

Finally, I enjoy the gift of an exceptionally supportive family. First and foremost, I thank my wife Dorah for her unconditional support and trust over the years. My children Verónica, Juan Carlos, and Hugo, have also shared the sacrifices and may have missed some of the attention they certainly deserve. I thank them for their confidence in me. My dear parents Norberto and Sara, and my brothers Norberto and Waldemar, have always been a source of inspiration and example to me. Finally, I want to express my gratitude to my father-in-law Félix, and to my dear late mother-in-law Luz. To all of you, please accept my deepest appreciation for your support.

Contents

List of Tables

List of Figures

Summary

This thesis investigates a number of image-adaptive, JPEG-compatible postfiltering and pre-post filtering methods designed to minimize the DCT *blocking* distortion. A pre-post filtering system uses inverse pair filters for high frequency preemphasis before encoding and high frequency deemphasis after decoding. The inverse pair pre-post filters are related by the inverse relationship $P(\omega) = 1/D(\omega)$, where P is the preemphasis filter and D is the deemphasis postfilter. A postfiltering system does not preemphasize the image before encoding. These techniques minimize the mean square error (MSE), improve the objective and subjective quality of low bit rate JPEG gray-scale images, and simultaneously enhance their perceptual visual quality. All the variants of the algorithms presented minimize the MSE below the level of baseline JPEG image compression, which is used as our comparison basis for similar bit rates. Convergence to a unique MMSE is possible for fixed quantization matrices, however, it cannot be guaranteed when image-adaptive quantization is jointly optimized under pre-post filtering.

We develop the theoretical basis of the *Estimated Spectrum Adaptive Postfilter* (ESAP) algorithm. ESAP is the main postfiltering algorithm used to minimize DCT blocking. ESAP utilizes either the default JPEG quantization table or image-adaptive DCT quantization matrices created in a preprocessing stage prior to image compression. At the decoder, the algorithm estimates 2-D pixel-adaptive bandwidths directly from the dequantized DCT coefficients to control a 2-D spatially-adaptive non-linear postfilter. Consistent with the human visual system tolerance to quantization errors in the high frequency regions, the algorithm performs directional filtering parallel to the edges and no filtering across the edges, subject to filter design constraints. Postfiltered images show minimal blurring of their true edges while blocking is significantly removed. ESAP relies on a DFT analysis of the DCT and is compliant with the coded stream syntax of the Independent JPEG Group (IJG) Version 5b Software.

Additionally, this thesis explores several other variants of the Estimated Spectrum Adaptive Postfilter applied to non-DCT coders such as vector quantization (VQ), subband coders (SBC), and Projection Onto Convex Sets (POCS). We compare the performance of these basic image coding methods against the same extended coders used with image preprocessing, ESAP postprocessing, or both, in a coder-compliant manner. These methods may or may not incorporate image-adaptive quantization and pre-post filtering. We also extend the concepts of the *dbx* audio noise reduction systems to model and demonstrate an *Iterative Pre-post Filter* (IPF). The IPF is applied to JPEG and to Set Partitioning In Hierarchical Trees (SPIHT) octave-band subband coders. We also process JPEG color images, synthetic aperture radar (SAR) images, and image sequences.

Typical PSNR improvement depends on the image, the encoding method, and the bit rate, and can range between 0.5–3.2 dB over baseline JPEG for 512×512 8-BPP gray-scale images. A comparison including all the treated techniques is presented at the conclusion of the thesis.

CHAPTER 1

Introduction

The Joint Photographic Expert Group (JPEG) image compression standard has become the most widely used international format for image compression and display among all the major computer applications. Most of JPEG's predominance stems from its ability to retain a relatively high image quality at low bit rates in the range of 0.5 to 1.0 BPP and its platform-independent format exchangeability. These were the design goals of the standard, and certainly, they have been achieved.

One typical application that makes extensive use of JPEG is Internet World Wide Web browsing. With the exponential growth of many web services, the ever-increasing need to push more image data through relatively slow 28–56 Kbps analog line Internet connections and the need to alleviate highly congested high-data-rate packet networks we need to both increase the speed of the connections and also improve the efficiency of the image coding algorithms. Given the wide acceptance of the JPEG standard, this thesis proposes various methods that could externally enhance its DCT coding performance to levels approaching the latest subband coders. We preprocess and postprocess the images in a manner that enhances the image quality even at a lower bit rate while still maintaining full JPEG compliance.

We believe that JPEG will continue to be one of the main image compression formats in the foreseeable future. Therefore, it is worth considering various alternatives that could increase its performance beyond its current limits. Additionally, these techniques could permeate into other applications such as low bit rate video teleconferencing and MPEG video coding and could improve their burgeoning presence in the expanding multimedia market.

This thesis investigates a number of image-adaptive JPEG-compatible postfiltering and pre-post filtering methods designed to minimize DCT *blocking* artifacts. The blocking distortion is the most significant limitation found in DCT-based still image coding systems. At low bit rates, DCT coders may exhibit major blocking artifacts and loss of quality. Blocking decreases the quality of the reproduction and could render the image or sequence useless beyond certain compression ratios. It arises from the artificial independent coding of 8×8 *minimum coded unit* (MCU) subimages which comprise the larger image.

Many mitigation techniques have been proposed as early as 1980. We survey a number of them, propose several new alternative blocking reduction algorithms and make some performance comparisons. All the proposed algorithms are modular and *external* to the JPEG still image compression standard. We ensure full JPEG compatibility by performing external image preprocessing, postprocessing, postfiltering, or pre-post filtering without altering the JPEG syntax to guarantee universal coder compatibility. These algorithms intend to improve both the objective and subjective quality of low bit rate gray-scale images at rates as low as 0.25 BPP. We model and test a number of the algorithms thoroughly and present broad objective and subjective assessments for each one.

The main building block of this thesis is a pixel-adaptive nonlinear postfilter named the *Estimated Spectrum Adaptive Postfilter* (ESAP) algorithm. ESAP's main objective is to enhance JPEG images by reusing the information stored in the decoded DCT coefficients of adjacent blocks. By interpolating this already available block-independent frequency domain information and reapplying it to the decoded images, we are able to adaptively postfilter the decoded images to improve their quality. This is done again in a JPEG-compliant manner without increasing the nominal bit-rate with large overhead. ESAP can be used in conjunction with JPEG adaptive quantization image preprocessing and pre-post filtering. Additionally, ESAP is extended to JPEG color images, gray-scale image sequences, applied to synthetic aperture radar (SAR) images, and finally, extended to subband coding aiming at improving the objective and subjective quality of these images while keeping the bit rate constant.

For reference purposes, the natural image examples to be presented are 512×512 or 256×256 8-BPP gray-scale images coded at 0.25, 0.5 and 1.0 BPP. The SAR images are 1008×1008 gray-scale images also coded at 0.25, 0.5 and 1.0 BPP.

Chapter 2 presents some background algorithms and image coding methods that will aid in understanding the forthcoming chapters of this thesis. Chapter 3 lays out the foundation of the ESAP algorithm. Chapter 4 discusses a number of postfiltering applications of ESAP to JPEG and non-JPEG images. Chapter 5 addresses image-adaptive pre-post filtering. This chapter presents the theory and experimental results of the *Iterative Pre-post Filter* (IPF) algorithm. Chapter 6 explores the idea of postfiltering and pre-post filtering of SBC images. Finally, we conclude this work in Chapter 7 stating the contributions and findings of this research. We propose some possible areas for future investigation.

CHAPTER 2

Background

The Joint Photographic Expert Group (JPEG) Standard [1, 2, 3, 4] has become the image compression format of choice among all major computer platforms. The advent of the World Wide Web [5] has made JPEG the most used lossy image compression standard. The latest and all future versions of the most popular web browsers support the JPEG *progressive* mode. This mode was specifically designed for low bit rate channels. Browsers initially display the text and then proceed to load a coarse version of any images in the hypertext document. Then, these images are progressively updated giving the viewer the illusion of a faster download.

The Telecommunications Act of 1996 [6], which deregulates telephone, cable, and broadcast television, strongly encourages DCT and SBC-based systems. It standardizes high definition digital television (HDTV) [7], an emerging DCT-based television standard that will eventually supersede the National Television Standard Committee (NTSC) analog television. The current DCT standards will certainly influence how much openness, scalability, and interoperability is achieved in the future, thus the importance of overcoming some of the DCT limitations. It is the intent of this thesis to offer some postfiltering and pre-post filtering image-enhancement algorithms that could help to promote widespread acceptance of DCT-based systems.

In the following sections, we present some definitions and background about the algorithms whose decoded images were processed with the Estimated Spectrum Adaptive Postfilter. This leads into the theory of ESAP and its present applications. ESAP is then followed by the Iterative Pre-post Filter algorithm. Finally, we present the contributions of this work and suggest some areas for future research.

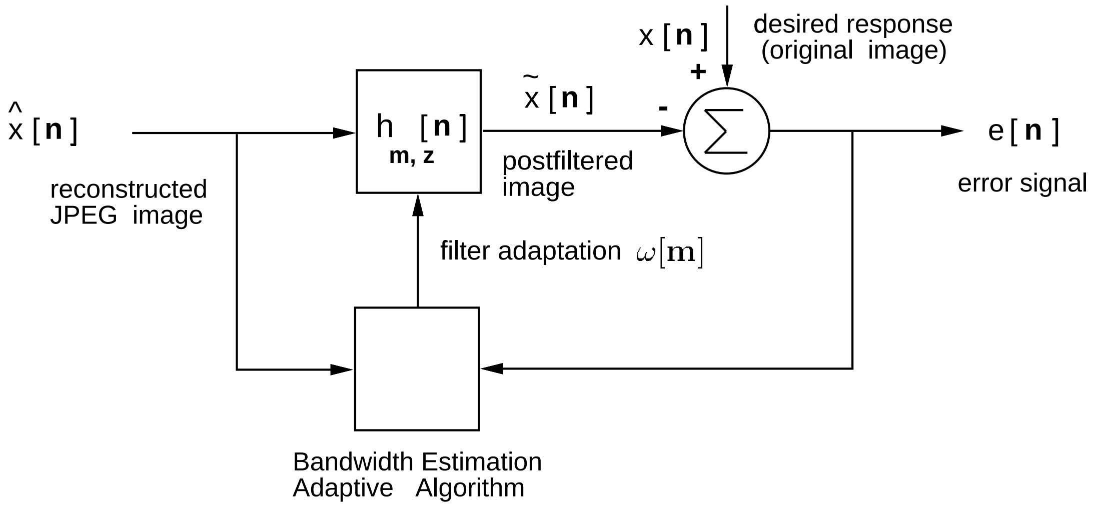


Figure 2.1: A 2-D adaptive filter or linear combiner.

2.1 Adaptive Filters

Let $\{m, n\}$ be discrete-time indexes. A 1-D *adaptive* filter [8] differs from a fixed filter in that its coefficients $h_m[n] = h[m-n] = h_{mn}$, vary as a function of the filter input $\hat{x}[n]$ at any discrete time m. Similarly, a 2-D adaptive filter with spatial indexes $\mathbf{m} = (m_1, m_2)$ and $\mathbf{n} = (n_1, n_2)$, differs from a 2-D fixed filter in that its coefficients $h_{\mathbf{m}}[\mathbf{n}] = h[\mathbf{m}-\mathbf{n}] = h_{\mathbf{mn}}$, vary with spatial displacement $\mathbf{m}$ as a function of the filter input $\hat{x}[n_1, n_2]$.

Referring to Figure 2.1, let $\hat{x}[\mathbf{n}]$ be a decoded JPEG image with its blocking artifacts caused by DCT coefficient quantization noise. Let a 2-D noncausal FIR adaptive lowpass filter $h_{\mathbf{m}}[\mathbf{n}]$, placed in a closed-loop configuration as a linear combiner [9], adapt its 2-D cutoff frequency $\omega[\mathbf{m}]$ as a function of the local bandwidth of the reconstructed signal $\hat{x}[\mathbf{n}]$ measured in a neighborhood $R^2_{\mathbf{m}} = [n_1 - m_1 \pm M/2, n_2 - m_2 \pm M/2]$, with M even. Let us assume that this local bandwidth can be estimated from the DCT of $\hat{x}[\mathbf{n}]$ over $R^2_{\mathbf{m}}$. At the encoder, the local pixel squared error is

$$e^2[\mathbf{n}] = (x[\mathbf{n}] - \tilde{x}[\mathbf{n}])^2. \tag{2.1}$$

Let $x \triangleq x[\mathbf{n}]$, $e \triangleq e[\mathbf{n}]$ and reshape $\hat{x}[\mathbf{n}]$ and $h_{\mathbf{m}}[\mathbf{n}]$ into the column vectors

$$\hat{\mathbf{X}}_{\mathbf{n}} \triangleq \begin{pmatrix} \hat{x}[\mathbf{n}_0] \\ \hat{x}[\mathbf{n}_1] \\ \vdots \\ \hat{x}[\mathbf{n}_{M\times M-1}] \end{pmatrix} \qquad \mathbf{H}_{\mathbf{mn}} \triangleq \begin{pmatrix} h_{\mathbf{m}}[\mathbf{n}_0] \\ h_{\mathbf{m}}[\mathbf{n}_1] \\ \vdots \\ h_{\mathbf{m}}[\mathbf{n}_{M\times M-1}] \end{pmatrix}. \tag{2.2}$$

By substituting (2.2) into (2.1) we obtain

$$\begin{aligned} e^2 &= (x - \hat{\mathbf{X}}_{\mathbf{n}}\mathbf{H}^T_{\mathbf{mn}})^2 \\ &= x^2 - 2x\hat{\mathbf{X}}_{\mathbf{n}}\mathbf{H}^T_{\mathbf{mn}} + \mathbf{H}^T_{\mathbf{mn}}\hat{\mathbf{X}}_{\mathbf{n}}\hat{\mathbf{X}}^T_{\mathbf{n}}\mathbf{H}_{\mathbf{mn}}. \end{aligned} \tag{2.3}$$

To obtain the MSE, we observe that x and $\hat{\mathbf{X}}$ are stationary signals and then take the expected value of (2.3)

$$\begin{aligned} \text{MSE} \ \triangleq \mu \ &= E\{e^2\} \\ &= E\{x^2 - 2x\hat{\mathbf{X}}_{\mathbf{n}}\mathbf{H}^T_{\mathbf{mn}} + \mathbf{H}^T_{\mathbf{mn}}\hat{\mathbf{X}}_{\mathbf{n}}\hat{\mathbf{X}}^T_{\mathbf{n}}\mathbf{H}_{\mathbf{mn}}\} \\ \mu \ &= E\{x^2\} - 2E\{x\hat{\mathbf{X}}_{\mathbf{n}}\}\mathbf{H}^T_{\mathbf{mn}} + \mathbf{H}^T_{\mathbf{mn}}E\{\hat{\mathbf{X}}_{\mathbf{n}}\hat{\mathbf{X}}^T_{\mathbf{n}}\}\mathbf{H}_{\mathbf{mn}}. \end{aligned} \tag{2.4}$$

The filter coefficients $\mathbf{H}_{\mathbf{mn}}$ are arbitrary deterministic constants for each $\mathbf{m}$. Therefore they can be placed outside the averaging operation. It is clear from (2.4) that μ is a *quadratic* function of the adaptive filter coefficients $\mathbf{H}_{\mathbf{mn}}$ when the input image x and its reconstruction $\hat{x}$ are stationary. The MSE is a summation of quadratic and linear terms over $\mathbf{m}$; therefore μ is quadratic. This becomes more explicit if we reformulate the MSE as

$$\begin{aligned} \mu &= E\{e^2\} \\ &= \frac{1}{N^2}\left(\sum_{\mathbf{m}=0}^{N^2-1} x^2_{\mathbf{m}} - 2\sum_{\mathbf{m}=0}^{N^2-1}\sum_{\mathbf{l}=0}^{M^2-1} x_{\mathbf{m}}\hat{x}_{\mathbf{ml}}h_{\mathbf{ml}} + \sum_{\mathbf{m}=0}^{N^2-1}\sum_{\mathbf{l}=0}^{M^2-1}\sum_{\mathbf{j}=0}^{M^2-1} h_{\mathbf{ml}}\hat{x}_{\mathbf{ml}}\hat{x}_{\mathbf{mj}}h_{\mathbf{mj}}\right) \end{aligned} \tag{2.5}$$

The triple summation in (2.5) emphasizes the quadratic nature of μ as a function of the adaptive filter coefficients $h_{\mathbf{mn}}$. Because of its quadratic nature, the MSE function has a *hyperparaboloid* shape in the number of parameters used by the adaptation algorithm. It must be upward concave and consequently has a *unique* minimum. As we will see, this is indeed the case for ESAP, which searches for the minimum mean square error (MMSE) or equivalently the peak signal-to-noise ratio (PSNR) as a function of two or four adaptive parameters. Once the MMSE point is found, it is transmitted to the decoder as side information.

In Chapter 5 we will find that the MMSE is no longer a unique minimum once we introduce JPEG image-adaptive coefficient quantization. The same is true for the SBC systems of Chapter 6. The MSE is approximately smooth but it can converge to local minima (Figure 6.2). It is well known that integer quantization is a nonlinear process. However, this nonlinear response is still sufficiently small to make the JPEG-IPF and the SBC-IPF algorithms useful despite the nonquadratic response of the MSE search surface.

2.2 Audio Noise Reduction Systems

The two most well known audio noise reduction systems are the *Dolby* and *dbx* audio filters [10, 11]. Although the dbx system was commercially dormant after the introduction of the Dolby A system in 1966, it has again reappeared in several computer applications such as PC soundcards. The dbx system is very useful in our analysis since it is the basis for Chapters 5 and 6 of this research.

The basic idea behind audio noise reduction is that low-level signals are dynamically boosted above the noise threshold during recording to mask the recording medium noise. At playback, the signal is dynamically attenuated, bringing it to its original dynamic range and simultaneously lowering the noise level. The dynamic boost is controlled by the signal's instantaneous amplitude, frequency content, and dynamic properties. Noise reduction systems can be static, dynamic, postfiltered, and pre-post filtered.

Tone controls are examples of static filters. A lowpass filter can reduce *hiss*, while a highpass filter reduces *hum.* Given an audio signal with a certain bandwidth, we can select how much of the low frequencies are retained using the bass control. The high frequencies are limited with the treble control. An equalizer is another example of a static bandpass filter. It selectively attenuates or boosts fixed frequency bands to suit the listener's preferences. A static filter does not change with variations in the input signal. An example of a dynamic filter is a limiter, which reacts only to loud signals.

2.2.1 Postfilters and Pre-post Filters

A *postfilter* only postprocesses the signal, producing a filtered approximation with lower noise. A postfilter can be a linear filter, a nonlinear filter, or a serial combination of linear and nonlinear filters. A *pre-post* filtering system performs both prefiltering and postfiltering with the aim of producing a signal with no alterations, while simultaneously reducing the noise. In a pre-post filtering system, saturation effects are minimized by *companding*, where signal *preemphasis* and *deemphasis* are performed before and after dynamic range compression and expansion, respectively. Pre-post linear filters are coupled inverse filters obeying a frequency domain relationship given by $P(\omega) = 1/D(\omega)$, where $P(\omega)$ is the preemphasis filter and $D(\omega)$ is the deemphasis postfilter. A pre-post nonlinear filter system contains three elements: a linear pre-post filter pair, and a nonlinear postfilter. In an audio noise reduction context, Figure 2.2 shows a simplified dbx pre-post filtering system. Figure 2.3 shows the dbx and a more complex Dolby B single-signal recording preemphasis characteristics at various input levels.

2.3 Lagrangian Optimization

The Lagrange multiplier method converts a constrained minimization problem into an unconstrained minimization problem. Let us assume that we wish to minimize a

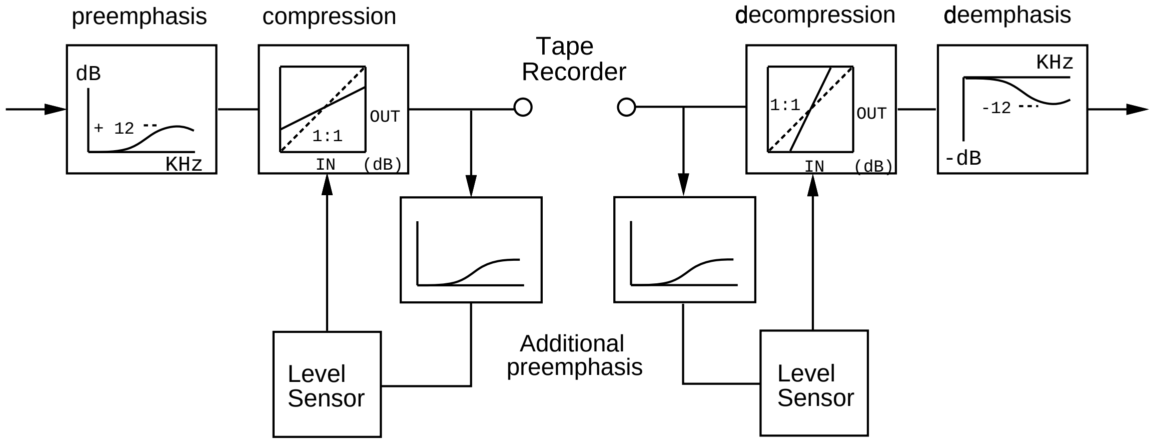


Figure 2.2: The *dbx* noise reduction system.

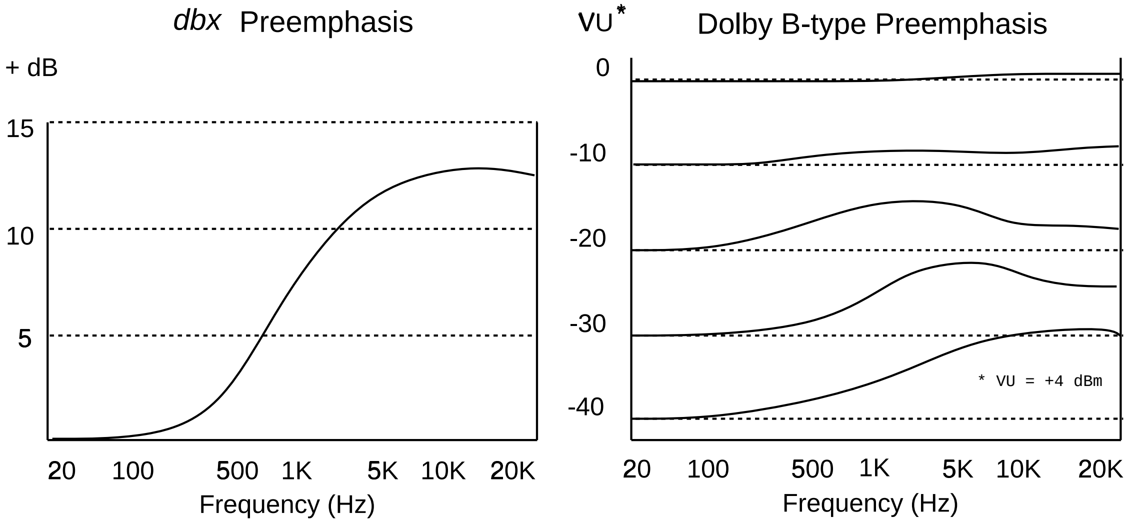


Figure 2.3: *dbx* and Dolby B-type preemphasis characteristics.

function $d(\cdot)$ subject to a constraint $r(\cdot) \leq R_0$. In other words

$$\min\{d(x_1, x_2, \ldots, x_m)\} \text{ subject to } r(x_1, x_2, \ldots, x_n) \leq R_0, m \neq n, m \cap n \neq \Phi, \quad (2.6)$$

where x_m and x_n are the independent variables of $d(\cdot)$ and $r(\cdot)$. A solution to (2.6) above can be obtained by minimizing the unconstrained cost function $J(\lambda)$

$$\min\{J(\lambda) = d(x_1, x_2, \ldots, x_m) + \lambda\, r(x_1, x_2, \ldots, x_n)\}. \quad (2.7)$$

By searching all the possible values of x_m and x_n for an appropriate set of scalar values of λ, solutions to (2.7) provide solutions to (2.6). If both $d(\cdot)$ and $r(\cdot)$ are multivariate monotonic functions of x_m and x_n respectively, it is then possible to perform a more efficient search instead of an exhaustive search. Even if $d(\cdot)$ is not completely monotonic and has local minima, as to be discussed in Sections 5.3 and 6.1, is is still possible to perform a suboptimal search that will produce very good results.

2.4 The JPEG Standard

The JPEG Standard defines four modes of operation: lossy sequential, progressive, hierarchical and the lossless sequential modes. Images are coded into minimum coded units or MCUs, which are multiples of 8×8-pixel blocks. The MCUs usually contain subsampled chrominance information and their data could also be interleaved. In the sequential mode, each MCU is encoded in a single pass.

In the progressive mode, each pass encodes part of the DCT coefficients. There are two ways to achieve this: *spectral selection* and *successive approximation.* In the spectral selection, the zigzag array of DCT shown in Figure 2.4 is grouped into bands. The lower bands are transmitted first; for example: (DC),(AC_{01}, AC_{10}), and then higher bands follow: $(AC_{20}, AC_{11}, AC_{02})$, etc., until all the coefficients have been transmitted. In the successive approximation mode, the most significant bits of each DCT coefficient are sent first and then the least significant bits follow. Successive

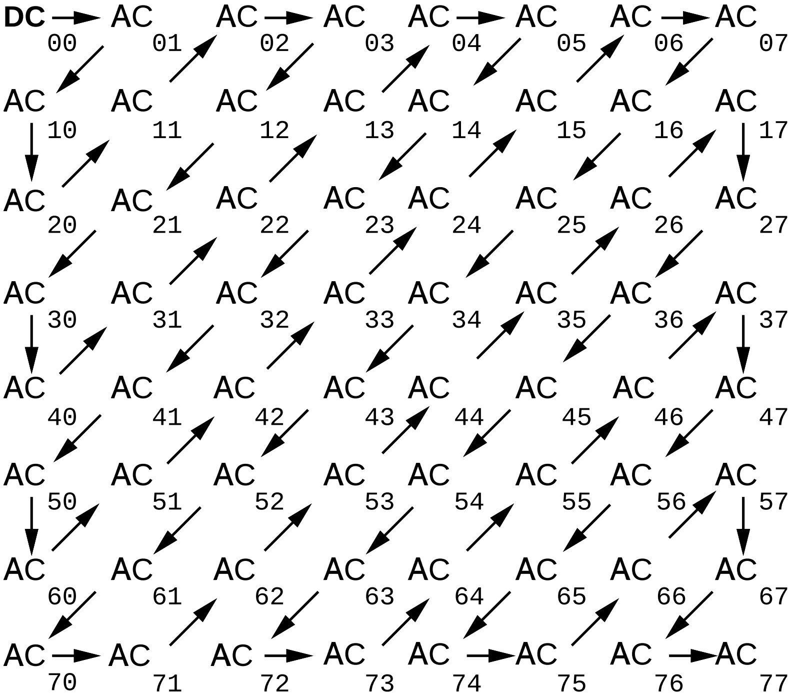


Figure 2.4: Zigzag reordering of DCT coefficients.

approximation displays better quality than spectral selection at comparable low bit rates.

The hierarchical mode increases the spatial resolution at each pass. The first pass of the smallest decimated image is either sequentially or progressively encoded. Afterwards, the output of each hierarchical stage is interpolated and used to predict the next increased resolution stage. A sequence of increased spatial resolution *differential* frames is computed and transmitted. At the decoder, the previous decimated image is upsampled and then the decoded differential frame is added to the upsampled prediction until the desired resolution is reached. This mode is also known as *pyramidal* coding. Finally, JPEG also has a lossless DPCM mode, which is summarized in Table 2.1.

In this thesis, we are mainly concerned with the *baseline* form of the sequential lossy mode. We now briefly discuss the main four JPEG subprocesses that are part of the baseline system, as well as the sequential and hierarchical modes. These are

Pixel Locations

c	b
a	x

Selection Value	Predictor x
0	no prediction
1	a
2	b
3	c
4	$a+b-c$
5	$a+(b-c)/2$
6	$b+(a-c)/2$
7	$(a+b)/2$

Table 2.1: Lossless JPEG predictors.

DCT transformation, DCT coefficient quantization, run length encoding, and entropy encoding.

DCT Transformation

JPEG uses separable Type-II forward and inverse DCT transforms [12]. The $N = 8$ forward DCT is

$$X(k_1,k_2) = \frac{1}{4}C(k_1)C(k_2)\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} x[n_1,n_2]\cos\frac{(2n_1+1)\pi k_1}{16}\cos\frac{(2n_2+1)\pi k_2}{16}, \tag{2.8}$$

and the inverse DCT is

$$x(n_1,n_2) = \frac{1}{4}\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} C(k_1)C(k_2)X[k_1,k_2]\cos\frac{(2n_1+1)\pi k_1}{16}\cos\frac{(2n_2+1)\pi k_2}{16}, \tag{2.9}$$

where

$$C(k_1),C(k_2) = \left\{\begin{array}{ll} \frac{1}{\sqrt{2}} & \text{for } k_1,k_2=0 \\ 1 & \text{otherwise} \end{array}\right\}.$$

JPEG does not mandate any specific DCT implementation. However, in order to obtain better real-time response, many JPEG coders substitute equations (2.8) and (2.9) with separable 1-D *fast* DCT transforms. For example, the Ligtenberg and Vetterlli algorithm shown in Figure 2.5 requires only 13 multiplications and 29 additions. In this figure, $C_k = \cos(k\pi/16), S_k = \sin(k\pi/16)$, and the rotations R_k are

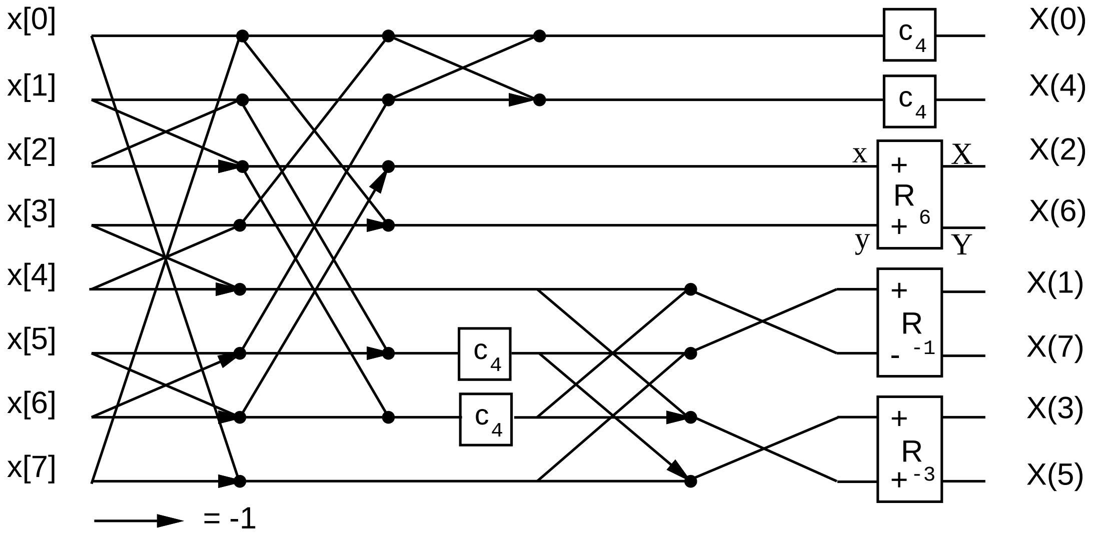


Figure 2.5: Flowgraph for fast DCT.

given by $X = C_k x + S_k y$ and $Y = -S_k x + C_k y$. Positive inputs to the rotation blocks are labelled "+". If the input is negative, the line is labelled with a "−", as shown in R_{-1}. The IJG Software Release 5b [4] uses a variant of this algorithm that requires 12 multiplications and 32 additions for the forward DCT [13]. On the other hand, a straightforward implementation would require 64 multiplications and 56 additions for each 1-D eight-point DCT. The inverse uses the Arai, Agui, and Nakajima's scaled DCT shown in Figure 4-8 of reference [3].

DCT Coefficient Quantization

JPEG provides two example DCT quantization tables, one for luminance and another for chrominance (Table 2.2). These values represent visibility thresholds for each DCT basis function. When the tables are divided by two, the reconstructed images are indistinguishable from the original. These tables are not optimized for any image in particular. They are based on the Human Visual System (HVS) spatial frequency response.

$$Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 27 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

$$C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Table 2.2: Luminance and chrominance quantization tables.

Run Length Coding

Once each block is DC level shifted, DCT transformed and quantized, the AC coefficients are reordered in a zigzag fashion. DC coefficients are differentially encoded based on the previous DC value. Low-frequency coefficients are likely to be nonzero, while high-frequency coefficients are almost always zero. This creates long runs of zero coefficients that can be *run length coded* (RLC). RLC is an intermediate representation before entropy coding is performed. RLC uses the symbols

Coefficients	**Symbol-1**	**Symbol-2**
AC	(RUNLENGTH,SIZE)	(AMPLITUDE)
DC	(SIZE)	(AMPLITUDE)

.

For the baseline option, RUNLENGTH varies from 0..15. Runlengths greater than 15 are extended by the Symbol-1 value (15,0), which is itself interpreted as a runlength of value 16. Thus, the maximum number of AC symbol-1 extensions is three. The value (0,0) is interpreted as end-of-block (EOB). SIZE gives the size of AMPLITUDE in bits for both AC and DC coefficients. AMPLITUDE is the amplitude of the

SIZE	AMPLITUDE
0	0
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1023..-512,512..1023
11	-2047..-1024,1024..2047

Table 2.3: Baseline variable length integers.

nonzero AC coefficients including sign, or the differential DC coefficient, based on the *variable length integer* (VLI) representation of Table 2.3. Symbols-1 are entropy encoded, while Symbols-2 are appended as VLIs to their corresponding Symbol-1 for transmission.

Entropy Coding

First-order entropy, or the average bit rate per information symbol, is defined as $H = -\sum_{k=1}^{M} p_k \log_2 p_k$ for symbols $x_1, x_2, \ldots, x_M$ with probalities $p_1, p_2, \ldots, p_M$. JPEG uses two methods for entropy coding: Huffman [14] and arithmetic coding [15, 16]. The Huffman codes are created by initially ordering the symbol probabilities p_k in decreasing order. Then, the two lowest-probability symbols are iteratively combined into a single symbol to create a new set of decreasing-order probabilities until only two symbols remain. A value of 0 is assigned to the higher probability symbol and a 1 to the lower. The code is created by proceeding backwards, decomposing probabilities, and appending 0 and 1 to the higher and lower decomposed probabilities, respectively.

Arithmetic coding does not require integer-length codes. Therefore, it can provide slightly better compression than Huffman coding. Its basic idea is to represent a symbol sequence x_n by a subinterval in the unit interval [0,1) based on symbol probabilities p_n, known to both coder and decoder, either as fixed probabilities or by adaptation. Starting with the initial conditions $n = 0, a_0 = 0, b_0 = 1$, and the $I_0 = [0, 1)$ interval, iteratively subdivide the previous interval I_{n-1} into two subintervals proportional to the new symbol probability having endpoints

$$a_n = \left\{ \begin{array}{cc} a_{n-1} & ; x_n = 0 \\ a_{n-1} + q(b_{n-1} - a_{n-1}) & ; x_n = 1 \end{array} \right\}, \qquad (2.10)$$

and

$$b_n = \left\{ \begin{array}{cc} a_{n-1} + q(b_{n-1} - a_{n-1}) & ; x_n = 0 \\ b_{n-1} & ; x_n = 1 \end{array} \right\}, \qquad (2.11)$$

where q is the probability of $x_n = 0$ and $(1 - q)$ is the probability of $x_n = 1$. The process is unambiguously reversible. If we know where the interval I_n falls in the [0,1) interval, we can trace back the whole sequence x_0, x_1, x_{n-1} by replicating the encoder's unit interval subdivision at the decoder until the terminator symbol is found.

2.4.1 Cross-block Smoothing

Section K.8 of the JPEG Standard provides a simple method to reduce block-to-block discontinuities. Cross-block smoothing (CBS) predicts the first five lower-frequency AC coefficients of each block by fitting a 2-D quadratic surface to a 3 × 3 array of 8 × 8-pixel blocks. By requiring that the mean of the predicted quadratic polynomial over each block be equal to the original DC value of the block, after appropriate scaling, quantization, and rounding, the AC coefficient prediction becomes

$$\begin{aligned} QAC_{01} &= \frac{RQ_{01} + 36Q_{00}(QDC_4 - QDC_6)}{256Q_{01}} \\ QAC_{10} &= \frac{RQ_{10} + 36Q_{00}(QDC_2 - QDC_8)}{256Q_{10}} \\ QAC_{20} &= \frac{RQ_{20} + 9Q_{00}(QDC_2 + QDC_8 - 2QDC_5)}{256Q_{20}} \end{aligned} \qquad (2.12)$$

$$\begin{aligned} QAC_{11} &= \frac{RQ_{11} + 5Q_{00}((QDC_1 - QDC_3) - (QDC_7 - QDC_9))}{256Q_{11}} \\ QAC_{02} &= \frac{RQ_{02} + 9Q_{00}(QDC_4 + QDC_6 - 2QDC_5)}{256Q_{02}}. \end{aligned}$$

R is $+128$ for positive numerators and -128 for negative numerators. QAC_{mn} is the AC prediction for the current center block based on the 3×3 DC-coefficient array

$$\begin{matrix} DC_1 & DC_2 & DC_3 \\ DC_4 & DC_5 & DC_6 \\ DC_7 & DC_8 & DC_9. \end{matrix} \tag{2.13}$$

The AC prediction results presented in this thesis were obtained from the IJG JPEG Version 4a implementation [4].

2.4.2 Visibility of DCT Basis Functions

The visibility of the gray-scale digitized sampling structure has been addressed in [17, 18, 19, 20]. The visibility of the DCT blocking structure has been treated in [21, 22, 23, 24]. The Mannos' contrast modulation transfer function (MTF), which is based on extensive human visual system (HVS) psychovisual testing and modeling, has the form

$$H(f) = 2.6 \ (0.0192 + 0.144f) \ e^{-(0.114f)^{1.1}}. \tag{2.14}$$

This function has its peak value at $f = 8$ cycles/degree and assumes that the HVS is isotropic. Another MTF, the Van Nes and Bouman, is similar to Mannos' MTF but peaks between five and eight cycles/degree.

Based on their own MTF peaking at four cycles/degree, Chitprasert and Rao proposed a weighted DCT scheme that retains sharp edges and smoothes DCT blocking. Their application of the MTF to the DCT weights the half-cycle DCT coefficients c_{01} and c_{10} with unity weights, while the rest of the coefficients are weighted with fractional weights according to their visibility. To justify this selection, assume a standard 0.28 mm pixel pitch display and a viewing distance of 1.0 m. This gives 0.13

degrees/block or about eight blocks/degree and a sampling frequency of $f_s = 64$ samples/degree. Since we need two adjacent blocks to create a full cycle with either c_{01} or c_{10}, the lowest frequency cosine basis function has a frequency of four cycles/degree. This result is very close to the peak sensitivity of the Mannos, Van Nes-Bouman, or Chitprasert-Rao MTFs. The next lower frequency at 8 cycles/degree is obtained with either c_{02} or c_{20}, which is also very visible. Beyond c_{03} or c_{30}, we reach the -3 dB point at 12 cycles/degree. Therefore, the most visible basis functions are the first six in the DCT zigzag reordering. This also coincides with the AC blocking prediction of Section 2.4.1.

Watson has proposed an algorithm that visually optimizes DCT quantization tables for specific images [25]. Peterson, Ahumada, and Watson have developed an improved DCT quantization algorithm that can predict visibility thresholds of DCT coefficient quantization errors for RGB and YC_rC_b color spaces [26]. These algorithms, which are now in the public domain [27], take into consideration the interdependencies among the DCT coefficients, the effects of contrast masking, display resolution and viewing distance. Using their models, the user can design Q-tables that keep quantization errors below the visibility threshold for the given viewing conditions. Similar independent work on perceptual optimization has also been reported in [28].

2.4.3 Image-Adaptive DCT Coefficient Quantization

Wu and Gersho [29], Ratnakar and Livny [30] and separately, Crouse and Ramchandran [31, 32] approached the DCT coefficient quantization optimization problem from a different perspective. Instead of adapting the Q-table solely to the HVS response, they optimize the Q-table in a *rate-distortion* (R-D) sense.

Wu uses a greedy algorithm. Starting with large Q-table step sizes, his algorithm decreases one Q-table entry at a time so that the ratio of MSE distortion decrease to bit-rate increase $\frac{\Delta D}{\Delta R}$ is maximized. Ratnakar's RD-OPT algorithm uses dynamic programming (DP) to generate optimal quantization tables, while Crouse's algorithm uses Lagrange multipliers in conjunction with DP. Ratnakar's DP algorithm optimizes a Q-table dependent rate $R(Q)$ against a Q-table dependent MSE distortion $D(Q)$. His procedure consists of computing the least-distortion path as each coefficient $Q(k,q), k = 0..63, q = 1..\texttt{MAXQ}$ is added based on the optimality principle: the least-distortion path to $D(k,q)$ must be the least distortion path to $D(k-1,q')$, plus the least distortion path from $D(k-1,q')$ to $D(k,q)$, where again, $q' = 1..\texttt{MAXQ}$.

Crouse and Ramchandran's adaptive quantization (AQ) algorithm minimizes the Lagrangian cost function

$$\min[J(\lambda) = D(T,Q) + \lambda R(T,Q,H)], \tag{2.15}$$

where λ is a quality factor, T is a set of 1 or 0 thresholding parameters that specify which coefficients are zeroed-out, and H is a Huffman table. Since each DCT block is independently coded, the algorithm uses DP to to minimize $J(\lambda)$ for each individual block and then adds the individual costs. Crouse uses a MSE distortion metric.

2.4.4 Joint Optimization

Crouse and Ramchandran went a step further by performing joint optimization (JO) of the Q-table, the coefficient thresholding parameters T, and the customized Huffman table. An iterative algorithm (i) optimizes $J(\lambda)$ as a function of Q, with T and H constant, based on some initial conditions. It then (ii) optimizes $J(\lambda)$ as a function of T, with Q and H constant and (iii) finally optimizes $J(\lambda)$ as a function of H with the previous computed Q and T. It then returns to (i) until convergence is met, i.e., when $J(\lambda)$ change is minimal. For the purposes of this thesis, we refer to the two versions of Crouse and Ramchandran's algorithms as adaptive quantization (AQ) and joint optimization (JO).

2.5 Canny Edge Detectors

We utilize Canny edge detectors [33] to detect image edges using the public-domain algorithm provided by Böe [34]. Canny edge detectors have three important characteristics.

1. **Good detection**. This refers to the detector's ability to mark real edges and reject noise as an edge. It is equivalent to maximizing the signal-to-noise ratio and is given by
$$SNR = \frac{\left|\int_{-W}^{W} G(-x) f(x) dx\right|}{n_o \sqrt{\int_{-W}^{W} f^2(x) dx}}, \tag{2.16}$$
where $f(x)$ is the impulse response of the FIR filter, bounded by [-W,W], $G(x)$ is the edge, and n_0^2 is the mean-squared noise amplitude per unit length.

2. **Good localization**. The detected center of an edge should be as close as possible to the true center. Localization is measured by
$$LOC = \frac{\left|\int_{-W}^{W} G'(-x) f'(x) dx\right|}{n_o \sqrt{\int_{-W}^{W} f'^2(x) dx}}. \tag{2.17}$$

3. **Eliminate multiple responses**. The edge detector should have no more than one response to a single step in the vicinity of an edge. Multiple responses can be minimized if the mean distance between the peaks of the filter's noise response are approximately the same width as the filter's response to a step.

Canny found that a filter whose impulse response is the first derivative of the Gaussian $f(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$ is a good edge detector that reasonably satisfies the above three requirements. It also matches very closely the optimal step detector that he found by numerical optimization.

2.6 Vector Quantization

Vector quantization (VQ), like the DCT, is a block-based coding technique. However, contrary to JPEG, VQ is *asymmetrical*, i.e., the encoder's computational complexity is much higher than the decoder's complexity, which usually is a lookup table with complexity $O(N)$. A vector quantizer Q of size N and dimension k maps a point in the k-dimensional Euclidean space $\mathcal{R}^k$ into a finite set $\mathcal{C}$ containing N output *code vectors* [35]. A vector quantizer matches its vector input to one of its predetermined codewords. Its performance is usually measured by the squared error or Euclidean distance

$$d(\mathbf{X}, \hat{\mathbf{X}}) = ||\mathbf{X} - \hat{\mathbf{X}}||^2 = \sum_{i=1}^{k} |X_i - \hat{X}_i|^2. \tag{2.18}$$

There are many types of vector quantizers. In this thesis, we attempt to enhance the output images of two vector quantizers: the entropy-constrained residual vector quantization (EC-RVQ) [36] and the conditional entropy-constrained residual vector quantization (CEC-RVQ) [37]. A three-stage RVQ encoder and decoder is shown in Figure 2.6. RVQ, also known as *multistage* or *cascaded* VQ, generates its output as the direct sum of the successive approximation reproduction from each stage. This arrangement reduces the overall computational complexity and memory requirements at the encoder since the codebook of each stage is smaller. However, the overall SNR will be lower compared with a single stage VQ encoder. The entropy-constrained qualification stems from the design of the encoder. It minimizes the entropy of the entropy-coded codewords instead of the length in bits of the variable-rate codewords of a pruned tree-structured vector quantizer (PTSVQ).

2.7 Projections Onto Convex Sets

Projection onto convex sets (POCS) is another recent technique employed for DCT blocking reduction [42, 43, 44, 45]. In a *convex* set, a line connecting any two points in the set is also a member of the set. Let S_1 and S_2 be two convex sets. Zakhor defined

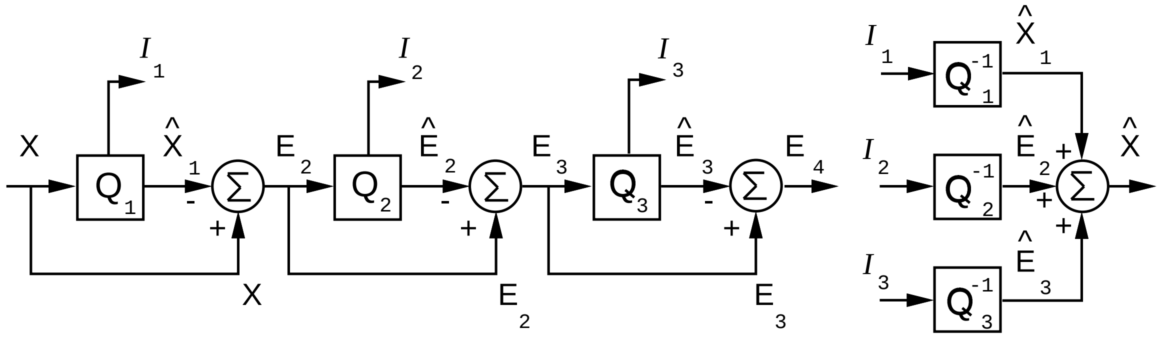


Figure 2.6: Residual vector quantizer encoder and decoder.

S_1 as the set of bandlimited (lowpass filtered) signals and S_2 as the set of images whose DCT coefficients are bounded by their respective quantization intervals. In her POCS algorithm, the first part of each iteration, performs (fixed) linear lowpass filtering to remove the high-frequency blocking discontinuities. The second part of each iteration recomputes the DCT for the filtered image and projects any coefficient outside its quantization range back onto its appropriate range. Basically, the POCS algorithm performs iterative *projections* between S_1 and S_2 until it converges to a common intersection point of the two sets. Su and Mersereau [46] substituted Zahkor's 3×3 linear LPF constraint by a linear shift-variant adaptive filter F, which was selected based on the 3×3 local pixel variance. In Section 4.4, we substitute the 3×3 filter with ESAP, with the intention of not only reducing the blocking, but sharpening the edges as well.

2.8 Subband Coders

A image subband coding (SBC) system consists of an analysis filter bank, a quantization and coding system for the subbands, and a synthesis filter bank [38, 39, 40]. Figure 2.7(a) shows an example of a tree-structured two-stage four-band subband coder. If the reconstructed image is identical to the input image, i.e., $\hat{x}[\mathbf{n}] = x[\mathbf{n} - \mathbf{n_0}]$, where $\mathbf{n_0}$ is the system delay, then the filter bank exhibits *perfect reconstruction.*

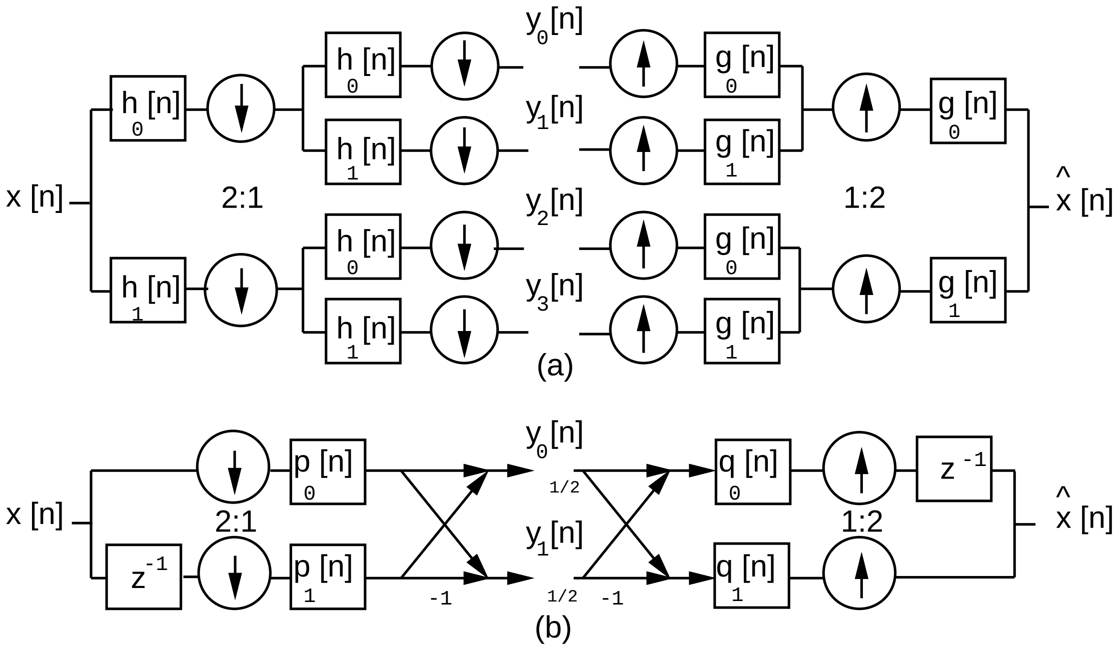


Figure 2.7: (a) A two-stage four-band subband coder. (b) One stage decomposition using polyphase filters.

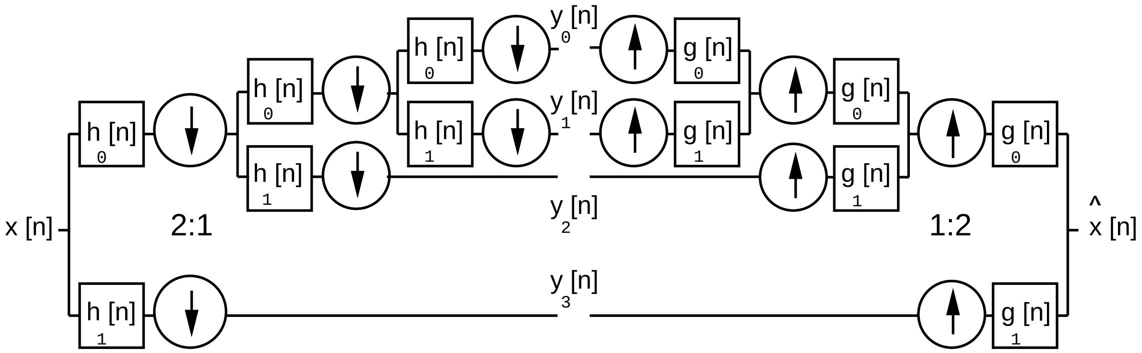


Figure 2.8: Diadic or octave-band tree-structured filter bank.

In addition, if all the bands can be downsampled by a factor of two and the total sampling rate of the input signal is equal to the total sampling rate of the subbands, then the filter bank is *critically down-sampled.* Filter banks are usually implemented using efficient *quadrature mirror filters* (QMF) with polyphase signal decomposition to reduce the computational complexity, as shown in Figure 2.7(b).

In Section 4.1, we compare the PSNR results of the *embedded zerotree wavelet* (EZW) algorithm [41] with ESAP. The EZW algorithm, is an octave-band tree-structured filter bank similar to the one shown in Figure 2.8. It hierarchically decomposes the lowpass band obtained from $h_0[n]$ into two subbands, while the high frequency signal obtained from $h_1[n]$ is not further subdivided (*diadic decomposition*).

2.9 The MPEG Standard

The Moving Picture Expert Group (MPEG) Standard [47, 48] has many features in common with national and international video-telephony [49, 50] and digital television standards [51]. MPEG is an open, interoperable, scalable[1] system designed to support a wide variety of video formats and compression rates. For instance, the MPEG-1 target rate is about 1.5 Mbps, which is comparable to VCR quality. On the other hand, HDTV uses MPEG-2 video coding at a target rate of 20–40 Mbps, which is equivalent to motion picture quality. Table 2.4 shows several MPEG formats. "I" means interlaced, "NI" means noninterlaced, and "P" means progressive scans.

MPEG defines three types of pictures: *intrapictures* (**I**), *predicted* pictures (**P**), and *bidirectional* predicted pictures (**B**). These pictures are assembled into *group of pictures* (GOP) resulting in sequences such as **I B B P B B P B B I B B P**..., etc. Intrapictures are basically stand-alone JPEG images. The **P** pictures can be forward or backward predicted. The prediction is based on block matching of 16 × 16-pixel motion-compensated *macroblocks.* **B** frames, also referred to as motion-

[1] Scalability refers to the user's ability to control the quality of the *received* image.

Format	Video Parameters	Aspect Ratio	Picture Rate (Hz)	Rate (Mbps)	Standard
SIF	352 × 240	4:3	30NI	1.2–3	MPEG-1
CCIR 601	720 × 486	16:9 & 4:3	60I 60P 30P 24P	5–10	MPEG-2
EDTV	960 × 486	16:9	60I 60P 30P 24P	7–15	MPEG-2
HDTV	640 × 480	4:3	60I 60P 30P 24P	20	MPEG-2
	1920 × 1080	16:9	60I 30P 24P	20–40	MPEG-2

Table 2.4: MPEG formats and rates.

compensated interpolated pictures, offer better performance than **P** pictures since the information is available from both "past" and "future" reference **I** frames. For **P** and **B** frames, MPEG computes a differential image resulting from the error between the motion-compensated prediction and a buffered original image. This prediction is DCT transformed, run-length and entropy encoded and then multiplexed along with the entropy-coded *motion vectors* and other overhead information such as quantizer step sizes. MPEG monitors a transmission buffer whose fullness is used to adjust the DCT quantizer step sizes.

2.10 Synthetic Aperture Radar

Synthetic Aperture Radar (SAR) imagery [71, 72, 73] is used for applications such as earth-resource mapping and military reconnaissance and targeting operations. These applications usually require imaging of broad areas at high spatial resolutions. In some instances, it is not possible to obtain this information in the visible spectrum. On the other hand, SAR is capable of imaging during night, day or inclement weather.

Some of SAR's uses include: sea ice monitoring, cartography, surface deformation detection, mineral exploration, autonomous navigation and guidance, glacier monitoring, crop production forecast, forest cover mapping, foliage and ground penetration, ocean wave spectra, urban planning, coastal erosion surveillance and disaster

Band	**Wavelenghth** (cm)
UHF	31–240
L	13–30
S	8–15
C	4–8
X	2.5–4
Ku	1.7–2.5
Ka	0.75–1.2

Table 2.5: Synthetic Aperture Radar bands.

monitoring. This includes forest fires, floods, volcanic eruptions and oil spills. SAR imaging complements other imaging systems in the infrared, visible and ultraviolet wavelengths.

The *radar* term itself is derived from the World War II phrase “radio detection and ranging.” SAR operating radio wavelengths range from a few centimeters to about two meters. The typical radar bands are shown in Table 2.5. These wavelengths cover a spectrum ranging from approximately 1 GHz to about 25 GHz. An overview of how SAR works and a brief tutorial on SAR image interpretation is offered below.

Typically, SAR imaging is conducted from an aircraft or an orbiting satellite such as LANDSAT or AVHRR. The 2-D SAR image is perpendicular to the aircraft velocity as shown in Figure 2.9. The *range* or along track dimension is a measure of the line-of-sight distance from the radar on the aircraft to the observed object. The SAR range is obtained by measuring the time delay from the transmission of a pulse to receiving the echo of that pulse from a target. Range resolution is obtained from the width of the transmitted pulse; the narrower the pulses, the finer the achievable range resolution. Typical SAR slant angles vary from 15 to 45 degrees.

The *azimuth* or cross track dimension is perpendicular to the range. One of the advantages of SAR imaging is its ability to produce fine azimuth resolution, which is otherwise only possible by using physically large antennas or with smaller antennas compensated by other methods. The large antenna is needed to focus the transmitted

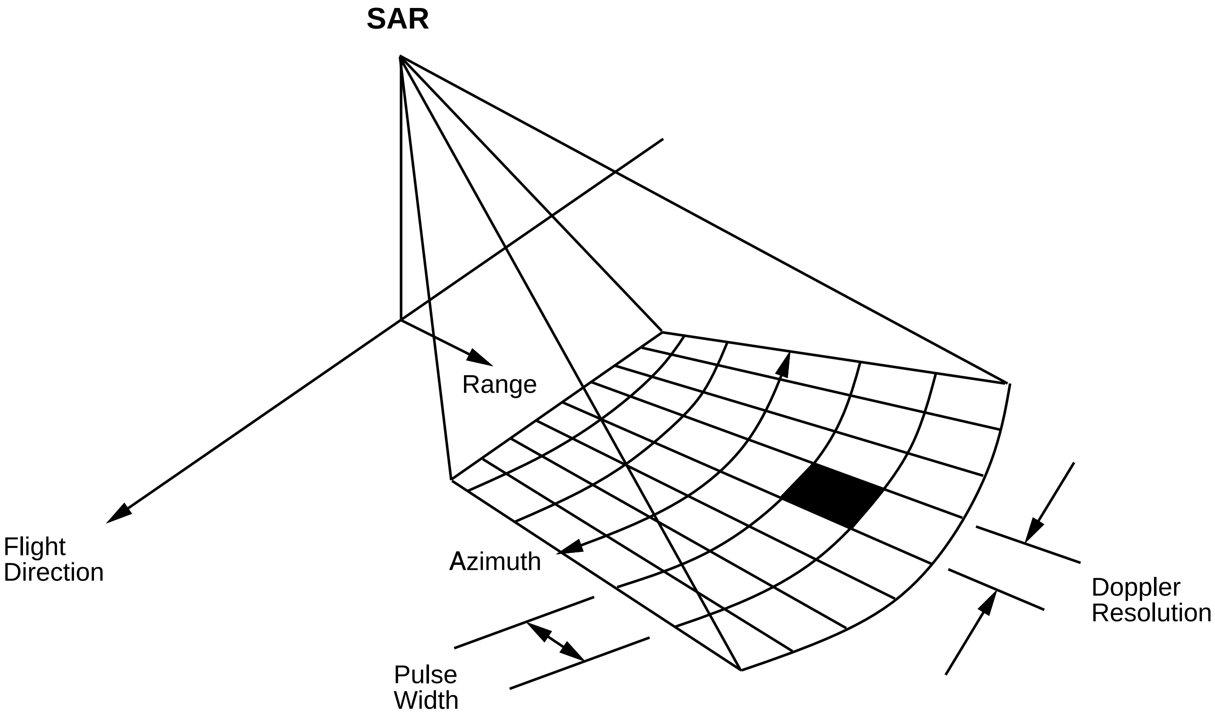


Figure 2.9: SAR data acquisition.

and received energy into a sharp beam to obtain the fine azimuth resolution. This physical phenomenon is similar to fine resolution in telescopes. These telescopes require large aperture mirrors or lenses to obtain fine imaging resolution. Given the relatively low operating radio frequencies for typical SAR systems in the electromagnetic spectrum, in order to obtain resolutions in the order of several meters it would be necessary to use an antenna that could be physically larger than the aircraft carrying it. Typical antenna lengths of several hundred meters might be required. An airborne radar can collect pulse echoes while flying several hundred meters and then process the data to model a physically longer antenna. The distance flown while collecting pulse reflections to synthesize the antenna is termed the *synthetic aperture*. SAR can achieve finer azimuth resolution than what is physically possible from a smaller antenna due to the synthetic aperture and its resulting narrow synthetic beamwidth.

SAR's fine azimuth resolution can also be explained through doppler frequency shifting. Objects ahead of the aircraft produce positive doppler offsets while objects behind produce negative offsets. The doppler frequency of the echoes along the flight path provides information about an object's position. Once the aircraft has flown the equivalent distance corresponding to the synthetic aperture, pulse echoes are resolved into doppler frequency components. An object's doppler frequency maps its azimuth position.

In practice, the transmission of short pulses to increase the range resolution is not feasible. Instead, SAR transmits longer pulses to decrease the transmitter's peak power requirements. This increases the computational load to determine the actual range. There are other factors that increase the computational complexity as well. For instance, azimuth resolution is affected by a target's range. The range to each cell on the synthetic aperture varies along the length of the synthetic aperture. There are also interdependencies between the range an azimuth that need to be decoupled prior to synthetic image generation.

There are a few guidelines for interpreting SAR images. In general black areas are associated with calm water or smooth surfaces that reflect away the incident pulses. Wet surfaces show more brightly than dry surfaces. Rough water shows up better than calm water, particularly if the wind-induced waves coincide with the wavelength of the incident radar signal. Mountains appear bright on the incident side and dim on the opposite side.

Backscatter is a function of the wavelength used. For example leaves and bushes are better shown at a wavelength of a few centimeters, say for instance the Ku band, than at a wavelength of one meter in the UHF band. UHF SAR wavelength is thus more appropriate for mapping larger objects such as geological formations. Human-made objects reflect the incident pulses very well, therefore they appear brighter than natural objects. Sharp corners in buildings, bridges and industrial complexes reflect extremely well, therefore these structure are clearly visible in the processed images.

In Section 4.9 we process a sample SAR image of the "Moving and Stationary Target Acquisition and Recognition" (MSTAR) Program taken at the Redstone Arsenal, Huntsville, Alabama by the Sandia National Laboratory [71] in September, 1995. These images used an X-band sensor operating at a 30 cm resolution with a 15 degree depression.

CHAPTER 3

The Estimated Spectrum Adaptive Postfilter

The ESAP algorithm extension to the baseline JPEG coder is shown in Figure 3.1. In the block diagram $x[\mathbf{n}]$ is the gray-scale input image, $\hat{X}_i(\mathbf{k})$ are the DCT coefficients of the transformed image, $\hat{x}[\mathbf{n}]$ is the decoded JPEG image and $\tilde{x}[\mathbf{n}]$ is the postfiltered image. The significance of the error signal $e(\sigma, f, t, w)$ is further explained in Section 3.4.

When preprocessing, to obtain better visual quality and a lower MSE, ESAP uses image-adaptive DCT quantization tables. This is not strictly necessary, but by doing so, we obtain about 1–2 dB PSNR improvement without increasing the bit rate. In postprocessing, ESAP estimates 2-D pixel-adaptive bandwidths directly from the dequantized DCT coefficients without incurring any additional side information. The postprocessing usually adds another 1 dB of improvement. The algorithm combines the pixel-adaptive bandwidths with directional Canny edge detectors to control a 2-D spatially-adaptive non-linear postfilter $h_{\mathbf{m}}[\mathbf{n}]$ that significantly reduces DCT blocking artifacts. The overhead required to transmit the MMSE postfilter parameters amounts to only two to four bytes. In our experiments, ESAP improved the PSNR up to 3.23 dB over baseline JPEG, yielding subjective improvement as well (see Section 4.1).

ESAP takes into consideration the HVS spatial frequency masking characteristics. Based on the HVS tolerance to quantization errors in the high-frequency regions, ESAP performs directional filtering *parallel* to the edges and no filtering *across* the edges, subject to filter design constraints. The low-frequency non-edge regions are

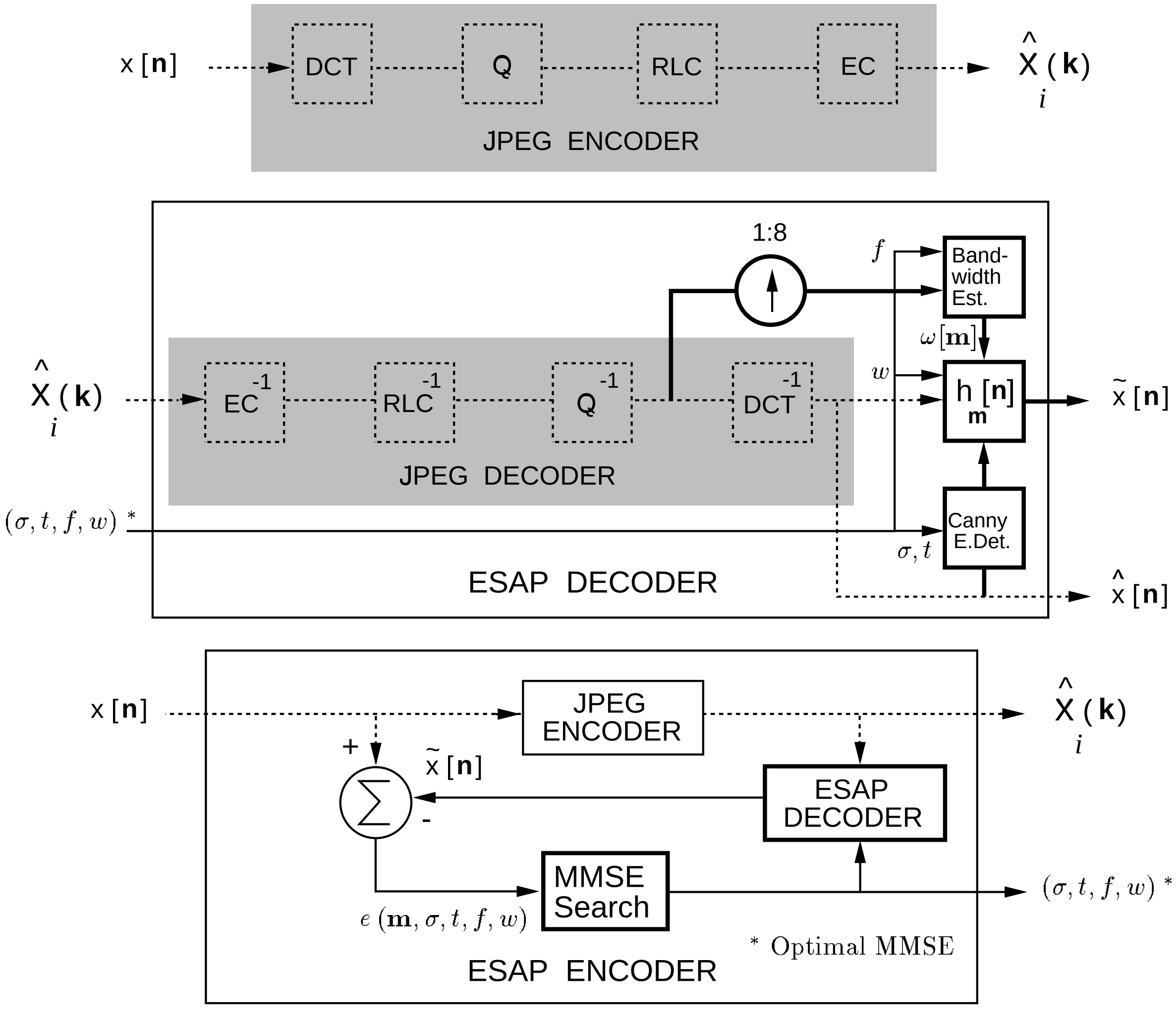


Figure 3.1: ESAP algorithm extension to baseline JPEG.

postfiltered with separable non-directional adaptive lowpass filters to minimize blocking and restore some of the image's natural smoothness. The edges are postfiltered with nonseparable directional lowpass filters. The filter's directionality helps to reduce the blocking along the local edge without significantly reducing the perceived fidelity across the edge, where high-frequency quantization discontinuity errors are masked by the HVS perception of the edge itself.

So far, we have mentioned several algorithms to reduce the DCT blocking distortion. Additionally, [52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63] attempted other methods with varying degrees of success. In [64, 65, 66], we proposed a solution to the DCT blocking limitation based on a frequency domain *analysis*. In the next section, we explain the Fourier transform foundation of the ESAP algorithm.

3.1 Fourier Transform Analysis of the DCT Basis Functions

Each DCT basis function has a Fourier transform whose waveform can be explained by the modulation theorem. In the 1-D case, this takes the form

$$x[n]w[n] \overset{\mathcal{F}}{\longleftrightarrow} \frac{1}{2\pi} X(\omega) \circledast W(\omega). \tag{3.1}$$

Making reference to Figure 3.2, observe that multiplication of a cosine function $f_j[n]$ by an eight-point rectangular window $w[n]$ in the time domain is equivalent to the circular convolution of a sinc-shaped rectangular window transform $W(\omega)$ with an ideal pair of impulses $\pi\delta(\omega \pm \omega_j)$ resulting in the $|F_j(\omega)|$ waveforms shown. Analysis of the discrete-time Fourier transform of each of the DCT basis functions [67], in conjunction with Figure 3.2, indicates that the ripples or sidelobes of the spectrum of each DCT basis function, shown as *dashed* lines, are the frequency representation of the DCT blocking at any particular spatial frequency. The window's width determines the main lobe frequency resolution and simultaneously introduces ripples.

Highest DCT Coeff. Present	Normalized 1-D Bandwidth
c_0	0.125π
c_1	0.250π
c_2	0.375π
c_3	0.500π
c_4	0.625π
c_5	0.750π
c_6	0.875π
c_7	1.0π

Table 3.1: Coefficient-block bandwidth relationship.

The ripples represent the *out-of-band* DCT blocking, while the main lobes contain the dominant *in-band* signal. If we neglect aliasing, then reduction of the DCT blocking is obtained by lowpass filtering the out-of-band sidelobes starting at a cutoff frequency ω_c determined by the highest-frequency nonzero DCT coefficient. This analysis is readily extensible to the 2-D case using a separable DCT. Figure 3.2 shows the ideal lowpass filters associated with each DCT basis function and Table 3.1 lists their cutoff frequencies.

Using Table 3.1, each block's bandwidth is found by inspecting the highest 2-D nonzero coefficient. Intermediate zero coefficients are neglected since they do not determine the block's bandwidth. For example, if the block's highest coefficient is c_{25}, then the *vertical* bandwidth is 0.375π and the *horizontal* bandwidth is 0.750π. The 2-D local bandwidth is centered in the middle of the block. This analysis generates two 64×64 bandwidth images from a 512×512 image. These are the non-interpolated *vertical* frequency (NIVF) image [Figure 3.3(a)] and the non-interpolated *horizontal* frequency (NIHF) image [Figure 3.3(c)]. In each of these, the gray level is proportional to the local horizontal or vertical bandwidth. Each NIF is subsequently 1:8 interpolated to obtain two 512×512 interpolated frequency (IF) images. Figures 3.3(b) and (d) show the interpolated vertical frequency (IVF) and the interpolated horizontal frequency (IHF), respectively. To properly filter the image boundaries, we need to symmetrically extend or replicate the IF images $\omega(\mathbf{m})$ and the decoded image $\hat{x}[\mathbf{n}]$.

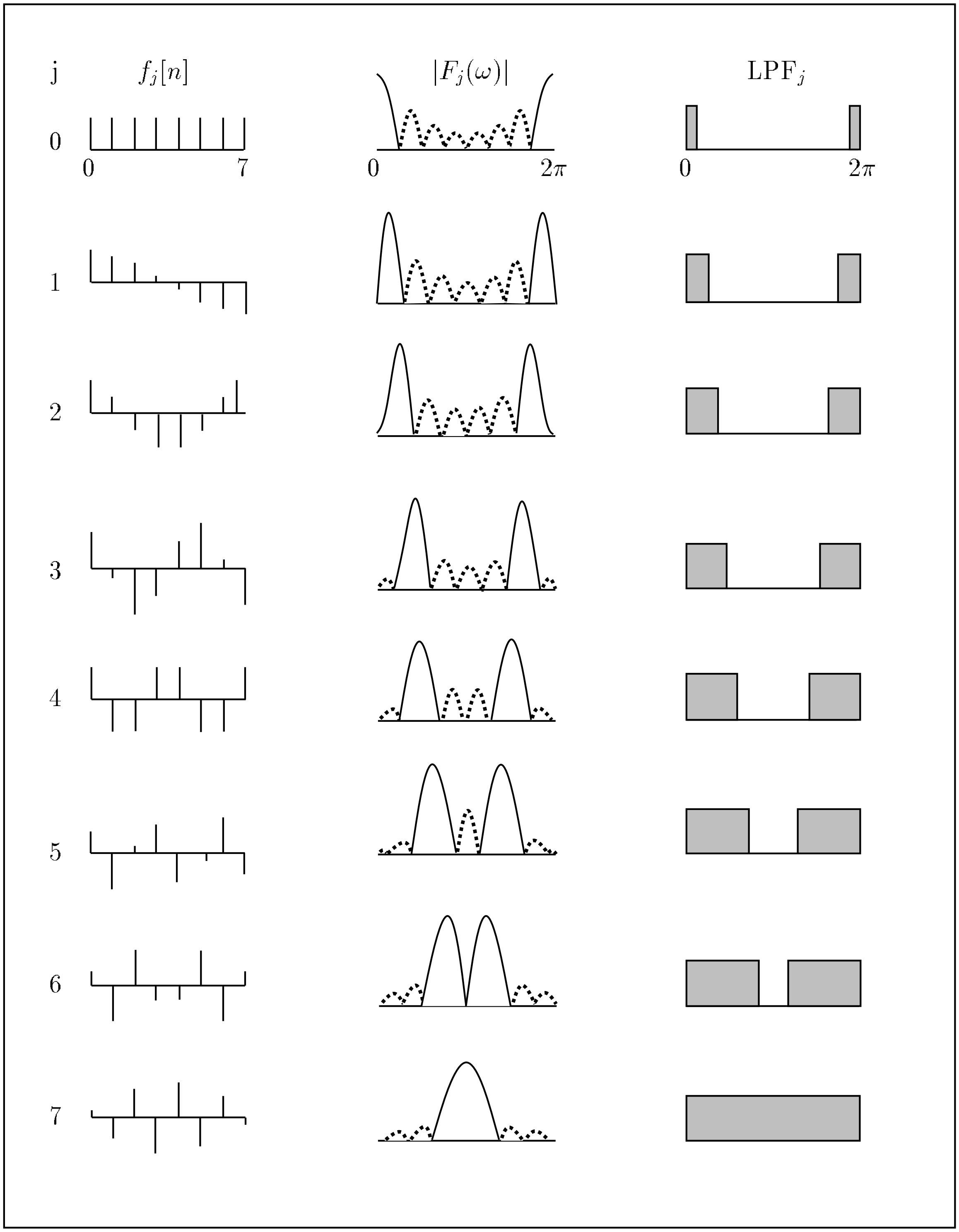
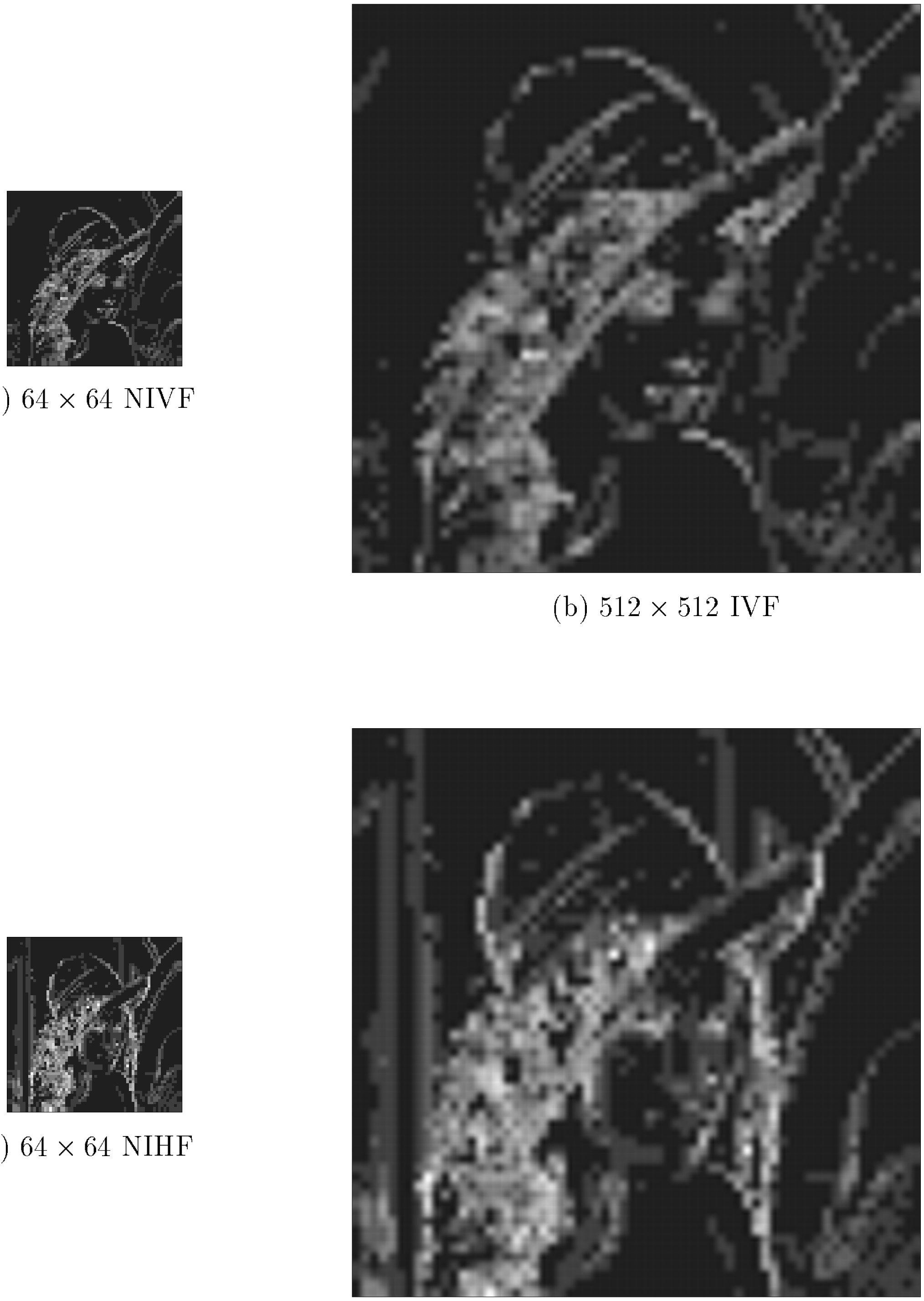


Figure 3.2: DCT basis functions $f_j[n]$, Fourier transforms $|F_j(\omega)|$, and associated LPFs.



(a) 64 × 64 NIVF

(b) 512 × 512 IVF

(c) 64 × 64 NIHF

(d) 512 × 512 IHF

Figure 3.3: Non-interpolated and interpolated frequency images.

This extends the decoded image by one MCU block on each side. For example, a 512×512 image increases to 528×528 pixels and the NIF images increase from 64×64 to 66×66 pixels. After the adaptive convolution is performed, the symmetrically extended blocks and NIF images serve no additional purpose and can be cleared.

3.2 Non-directional Filtering

ESAP smoothes the current pixel of the decoded image $\hat{x}[m_1, m_2]$ with a 2-D adaptive cutoff lowpass FIR Hamming filter $h_{m_1,m_2}[n_1, n_2]$ which can be directional or nondirectional. The filter's directionality is determined from the output of a Canny edge detector applied to the decoded image. The horizontal and vertical bandwidths are obtained from the IHF and IVF images, respectively. EDGE and NON-EDGE pixels are found by the following rule:

```
                /* Classify EDGE & NON-EDGE pixels */

   for (n1=0; n1<N; n1++)
      for (n2=0; n2<N; n2++)
         if (canny_mag[n1][n2] > T && sqrt(Bw1[n1][n2]*Bw1[n1][n2] +
                                            Bw2[n1][n2]*Bw2[n1][n2]) > F)
            edge[n1][n2] = TRUE;
         else
            edge[n1][n2] = FALSE;
```

`T` is a Canny edge magnitude threshold, `F` is a normalized 2-D frequency magnitude threshold, and `Bw` are IFs[1]. In other words, if a pixel is an EDGE in *both* the spatial domain *and* the frequency domain, then the pixel is declared a true image EDGE and it is directionally postfiltered. If a pixel's Canny magnitude is $<$ `T` but its 2-D bandwidth magnitude is $>$ `F`, then it is declared a TEXTURE pixel and filtered with an impulse $\delta[n_1, n_2]$. Otherwise, the NON-EDGE pixel is nondirectionally postfiltered.

[1] In Figure 3.1, we use the equivalent notation t, f and $\omega[\mathbf{m}]$ for `T`, `F` and `Bw` respectively. The parameters σ and w are explained in Section 3.4.

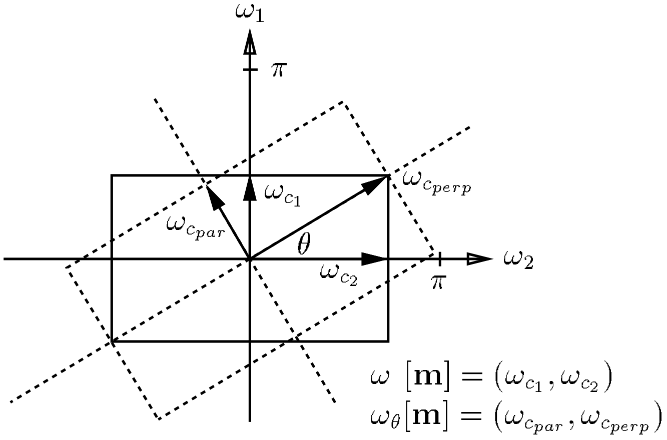

ω_1
π
$\omega_{c_{perp}}$
ω_{c_1}
$\omega_{c_{par}}$
θ
ω_2
ω_{c_2}
π
$\omega\ [\mathbf{m}] = (\omega_{c_1}, \omega_{c_2})$
$\omega_\theta[\mathbf{m}] = (\omega_{c_{par}}, \omega_{c_{perp}})$

Figure 3.4: Determination of rotated bandwidths $\omega_\theta[\mathbf{m}]$.

3.3 Directional Filtering

Once we have determined the angular orientation θ of an edge passing through a pixel $\hat{x}[\mathbf{m}]$ using the Canny edge detector, we can compute a *rotated* bandwidth $\omega_\theta[\mathbf{m}]$ from the original rectangular bandwidth estimation $\omega[\mathbf{m}]$. Let us look at Figure 3.4, which describes a typical block with a substantial amount of diagonal frequencies.

Observe that the magnitude of the highest 2-D frequency in the frequency plane corresponds to the *perpendicular* cutoff frequency across the edge and is given by

$$\omega_{perp} \approx \min(1.0\pi, \sqrt{\omega_{c_1}^2 + \omega_{c_2}^2}\ \pi). \tag{3.2}$$

To exploit the HVS high-frequency masking characteristics, we fix $\omega_{c_{perp}} = \pi$. The cutoff frequency *parallel* to the edge is approximately the lower of the vertical (ω_{c_1}) or horizontal (ω_{c_2}) DCT bandwidths:

$$\omega_{c_{par}} \approx \min(\omega_{c_1}, \omega_{c_2}). \tag{3.3}$$

Now, a noncausal 2-D rotated Hamming filter can be expressed as

$$h_\theta[\omega_{c_{par}}, \omega_{c_{perp}}, n_1, n_2] =$$

$$h_{\omega_{c_{par}}}[\sqrt{n_1^2+n_2^2}\sin(\theta + \tan^{-1}\frac{n_1}{n_2})]\ h_{\omega_{c_{perp}}}[\sqrt{n_1^2+n_2^2}\cos(\theta + \tan^{-1}\frac{n_1}{n_2})], \tag{3.4}$$

where θ is measured counterclockwise with respect to the horizontal axis n_2 (or ω_2) and the 1-D Hamming window LPF is given by

$$h_{\omega_c}[n] = \frac{\sin(\omega_c n)}{\pi n}\,[0.54 - 0.46\cos\frac{2\pi n}{M}] \qquad 0 \leq n \leq M. \tag{3.5}$$

For NON-EDGE pixels, the above expression simplifies to the separable filter

$$h[\omega_{c_1}, \omega_{c_2}, n_1, n_2] = h_{\omega_{c_1}}[n_1]\ h_{\omega_{c_2}}[n_2]. \tag{3.6}$$

The nondirectional 1-D filters of equation (3.6) are precomputed at program initialization and accessed as a lookup table during execution. ESAP's computational complexity is approximately $O((MN)^2)$ multiplications and additions for the estimated frequency interpolation and $O((MN)^2)$ additions, $O(N^2M^2/4)$ multiplications for the adaptive convolution of each image ($N \times N$ is the image size and $M \times M$ is the filter size).

Equations (3.2) and (3.3), and the model of Figure 3.4, are based on actual separable bandwidth measurements obtained from the quantized DCT coefficients of the rotated edges of an image, as explained in Section 3.1. Please also note that for both directional and nondirectional filtering, the pixel-adaptive nature of the IF images forces us to use spatially-adaptive convolution or equivalently, a linear combiner. In other words, the actual implementation *cannot* use frequency-domain filtering. Nevertheless, the analysis presented in Section 3.1 is useful in determining the adaptive filter's 2-D bandwidth, although the filtering operation is actually performed in the spatial domain.

3.4 MMSE Optimality of ESAP

3.4.1 Definitions

In reference to Figure 3.1 let

σ = Canny edge detector's Gaussian standard deviation *sigma* parameter (Section 2.5), usually in the range [0.5..2.5]. This controls the region of support of the detector.

t = Canny edge detector's edge strength *threshold*, [0..255] range.

f = Magnitude of ESAP's DCT *frequency* bandwidth, $[0.0..\sqrt{2}]\pi$ range. In conjunction with t, this classifies pixels into three categories: EDGE, NON-EDGE or TEXTURE.

w = Directional filter region of support $w \times w$ in pixels, $w = [2..16]$. Nondirectional filters have a fixed order of 17×17 pixels to cover four contiguous DCT blocks.

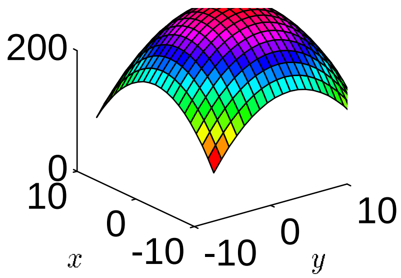
For example, these parameters can be quantized and packed in three bytes to represent the MMSE adaptive overhead information as follows:

$$\sigma_{\ 6\ bits} = [0.04,\, 0.08,\, \ldots,\, 2.56] \qquad t_{\ 8\ bits} = [0,\, 1,\, \ldots,\, 255]$$
$$f_{\ 7\ bits} = [0.011,\, 0.022,\, \ldots,\, 1.419] \qquad w_{\ 3\ bits} = [2,\, 4,\, \ldots,\, 16].$$

In our case, the MSE μ, or its logarithmic inverse, the PSNR $p = 10 \log \frac{255^2}{\mu}$, are both scalar multivariate functions of the arbitrary filter coefficients $h_{\mathbf{m}}[\mathbf{n}]$ (please refer to Section 2.1). In turn, these filter coefficients, of up to 17×17 support in this algorithm, are a vector function of the 4-tuple $\mathbf{z} = (\sigma, t, f, w)$, in addition to the pixel's 2-D coordinate $\mathbf{m}$. In expanded notation

$$\mu = f(x[\mathbf{n}],\ \hat{x}[\mathbf{n}],\ h_{\mathbf{m},\mathbf{z}}[\mathbf{n}]). \tag{3.7}$$

By controlling $\mathbf{z}$, we are indeed controlling the adaptive filter coefficients $h_{\mathbf{m},\mathbf{z}}[\mathbf{n}]$. Since μ is a quadratic MSE function of $h_{\mathbf{m},\mathbf{z}}[\mathbf{n}]$ with a unique minimum, then by inference, μ must also be a quadratic function of $\mathbf{z}$. Consequently, the MMSE $\mu(\mathbf{z}^*)$ has an optimal logarithmic inverse maximum PSNR $p(\mathbf{z}^*)$. We want to maximize the PSNR (the MSE variable $\mu(\mathbf{m}, \sigma, t, f, w)$ could have been minimized as well), therefore we choose to maximize the variable $p(\mathbf{m}, \sigma, t, f, w)$ by performing a steepest-ascent gradient search.



$g(x,y) = 200 - (x^2 + y^2)$

Figure 3.5: Inverted parabola PSNR analogy.

3.4.2 Graphical Interpretation

Consider the 2-D inverted parabola $g(x,y) = 200 - (x^2 + y^2)$ in Figure 3.5. If we slice g with a plane $x = x_i$ [or $y = y_i$], we always obtain a 1-D inverted parabola with a maximum $g(x_i, 0)$ [or $g(0, y_i)$.] The 2-D absolute maximum is the maximum of all the 1-D maxima — $g(0,0)$ in this example.

We can extrapolate this heuristic to the 4-D PSNR surface $p(\sigma, t, f, w)$, which has been evaluated for all square errors $e^2[\mathbf{m}]$. For each pair (σ_i, t_j), if *all* the resulting 2-D surfaces $p(\sigma_i, t_j, f, w)$ are downward monotonic for all possible (f, w) points in the 2-D region of support $R_2 = \{F \times W\}$, then there exists a maximum point $p(\mathbf{z}^*) = p(\sigma^*, t^*, f^*, w^*)$, which is the maximum of all the 2-D maxima $p(\sigma_i, t_j, f^*, m^*)$ over R_2. Figure 3.6 shows ESAP's behavior through two downward-monotonic surface examples obtained from experimental data of the "Lena" image compressed at 0.25 BPP. The next section—Matrix Analysis, presents an alternative *optional* explanation of the MMSE optimality of the algorithm.

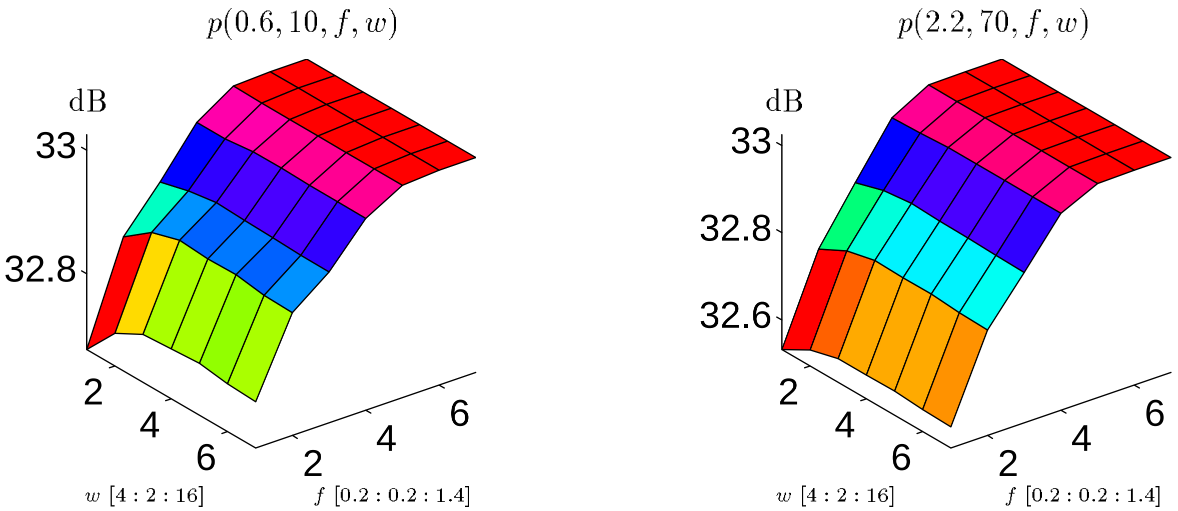


Figure 3.6: ESAP monotonic PSNR surfaces $p(\sigma_i, t_j, f, w)$.

3.4.3 Matrix Analysis

Another approach that could be used to verify ESAP's optimality is based on matrix analysis [68]. To maximize p, it is necessary that

$$\partial p/\partial \mathbf{z} = \begin{pmatrix} \partial p/\partial \sigma \\ \partial p/\partial t \\ \partial p/\partial f \\ \partial p/\partial w \end{pmatrix} = \mathbf{0}. \tag{3.8}$$

That is, p has to attain a stationary value $\mathbf{z}^* = (\sigma^*, t^*, f^*, w^*)$. Also, the square symmetric *Hessian* matrix

$$\mathbf{H} = -\partial\left(\frac{\partial p}{\partial \mathbf{z}}\right)\partial \mathbf{z}' = -\left.\begin{pmatrix} \frac{\partial^2 p}{\partial \sigma^2} & \frac{\partial^2 p}{\partial \sigma \partial t} & \frac{\partial^2 p}{\partial \sigma \partial f} & \frac{\partial^2 p}{\partial \sigma \partial w} \\ \frac{\partial^2 p}{\partial t \partial \sigma} & \frac{\partial^2 p}{\partial t^2} & \frac{\partial^2 p}{\partial t \partial f} & \frac{\partial^2 p}{\partial t \partial w} \\ \frac{\partial^2 p}{\partial f \partial \sigma} & \frac{\partial^2 p}{\partial f \partial t} & \frac{\partial^2 p}{\partial f^2} & \frac{\partial^2 p}{\partial f \partial w} \\ \frac{\partial^2 p}{\partial w \partial \sigma} & \frac{\partial^2 p}{\partial w \partial t} & \frac{\partial^2 p}{\partial w \partial f} & \frac{\partial^2 p}{\partial w^2} \end{pmatrix}\right|_{\mathbf{z}^*} \tag{3.9}$$

has to be positive-definite satisfying $\mathbf{z}^*$ above. If $\mathbf{H}$ is positive-semidefinite, then $\mathbf{z}^*$ is a *weak* maximum.

Since equations (3.8) and (3.9) operate on discrete values, they can be approximated by partial *difference* equations. Therefore,

$$\partial p/\partial \mathbf{z} \simeq \begin{pmatrix} p(\sigma,t,f,w) - p(\sigma-1,t,f,w) \\ p(\sigma,t,f,w) - p(\sigma,t-1,f,w) \\ p(\sigma,t,f,w) - p(\sigma,t,f-1,w) \\ p(\sigma,t,f,w) - p(\sigma,t,f,w-1) \end{pmatrix} \triangleq \begin{pmatrix} p(\sigma) - p(\sigma-1) \\ p(t) - p(t-1) \\ p(f) - p(f-1) \\ p(w) - p(w-1) \end{pmatrix} = \mathbf{0}. \tag{3.10}$$

For notational convenience, the variables not shown remain constant. **H** now becomes

$$\mathbf{H} \simeq - \begin{pmatrix} p(\sigma) - 2p(\sigma-1) + p(\sigma-2) & p(\sigma,t) - p(\sigma-1,t) - p(\sigma,t-1) + p(\sigma-1,t-1) & p(\sigma,f) - p(\sigma-1,f) - p(\sigma,f-1) + p(\sigma-1,f-1) & p(\sigma,w) - p(\sigma-1,w) - p(\sigma,w-1) + p(\sigma-1,w-1) \\ p(\sigma,t) - p(\sigma-1,t) - p(\sigma,t-1) + p(\sigma-1,t-1) & p(t) - 2p(t-1) + p(t-2) & p(t,f) - p(t-1,f) - p(t,f-1) + p(t-1,f-1) & p(t,w) - p(t-1,w) - p(t,w-1) + p(t-1,w-1) \\ p(\sigma,f) - p(\sigma-1,f) - p(\sigma,f-1) + p(\sigma-1,f-1) & p(t,f) - p(t-1,f) - p(t,f-1) + p(t-1,f-1) & p(f) - 2p(f-1) + p(f-2) & p(f,w) - p(f-1,w) - p(f,w-1) + p(f-1,w-1) \\ p(\sigma,w) - p(\sigma-1,w) - p(\sigma,w-1) + p(\sigma-1,w-1) & p(t,w) - p(t-1,w) - p(t,w-1) + p(t-1,w-1) & p(f,w) - p(f-1,w) - p(f,w-1) + p(f-1,w-1) & p(w) - 2p(w-1) + p(w-2) \end{pmatrix}. \tag{3.11}$$

For $\mathbf{z}^*$ to be a maximum, **H** has to be positive-definite or at least positive-semidefinite.

It is a well-known fact that the MSE is a hyperparaboloid function of its adaptive filter coefficients, and indirectly of the $\mathbf{z}$ variable, if $p(\mathbf{z})$ is a stationary signal (i.e. not affected by quantization, etc.) Therefore, it is not really necessary to implement a proof based on a numeric implementation of (3.10) and (3.11) above. However, for completeness, we outline the numerical proof algorithm below.

The algorithm would first find all the 4-D stationary points p_s in the region of support $R_4 = \{S \times T \times F \times W\}$ that satisfy (3.10). Secondly, it would compute $\mathbf{H}$ using (3.11) for each p_s. Finally, it would determine four eigenvalues $\lambda\ _{1,2,3,4}$ and four eigenvectors $u\ _{1,2,3,4}$ for each p_s by solving the 4×4 characteristic equation

$$(\mathbf{H} - \lambda\mathbf{I})\mathbf{U} = \mathbf{0}. \tag{3.12}$$

The 4-D point that satisfies (3.10) and whose eigenvalues λ are all *positive,* is the optimal maximum point $p(\mathbf{z}^*)$.

3.5 O((2D - 1)N) ESAP 4-D Fast Optimal Search

There are N^4 PSNR points p in a 4-D hyperspace region of support $R_4 = \{S \times T \times F \times W\}$, where S, T, F, and W represent N equally spaced positive coordinates on each Cartesian axis of the $\mathbf{z}$ space. Since the 4-D ESAP quadratic surface p is downward monotonic, it is unnecessary to use all the points in R_4. An optimal search can be performed to reduce the exhaustive search complexity from $O(N^D)$ to a worst case diagonal or lateral search of order $O(\lceil \frac{3^D}{2} \rceil N)$, where D is the number of dimensions in R. This order represents the neighboring points around the 4-D diagonal or a coordinate axis. For example, if N=10 and D=4, then the exhaustive search requires $O(10^4)$ or 10,000 postfiltering operations. The optimal 4-D full search would only require about $O(\lceil \frac{3^4}{2} \rceil 10) = 410$ postfilterings. At each step, we would compute an average of $\lceil \frac{3^D}{2} \rceil$ new points since half of those were already computed in the previous iteration.

We can further reduce the number of postfiltering operations to $O((2D-1)N)$ and still obtain optimal results if we use the two nearest neighboring points on each axis parallel to S, T, F, and W, instead of the complete D-dimensional neighborhood of a point. This is equivalent to searching only those hypercube points that intersect perpendicularly to the reference axes, plus the origin. In the case of R_4 with N=10, we can converge to the optimal $p(\mathbf{z}^*)$ in $O((2(4)-1)10) = 70$ operations, a reduction greater than two orders of magnitude with respect to an exhaustive search.

Most of the $\mathbf{z}^*$ search results to be presented in Chapter 4 were obtained by nested for-loop exhaustive searches that ran from hours to days on a SPARCStation 2 computer. However, when we used the two-nearest neighbors on each axis $O((2D-1)N)$ heuristic described above, we were able to converge to a MMSE solution in less than an hour, given proper initial conditions $(\sigma_o, t_o, f_o, w_o)$.

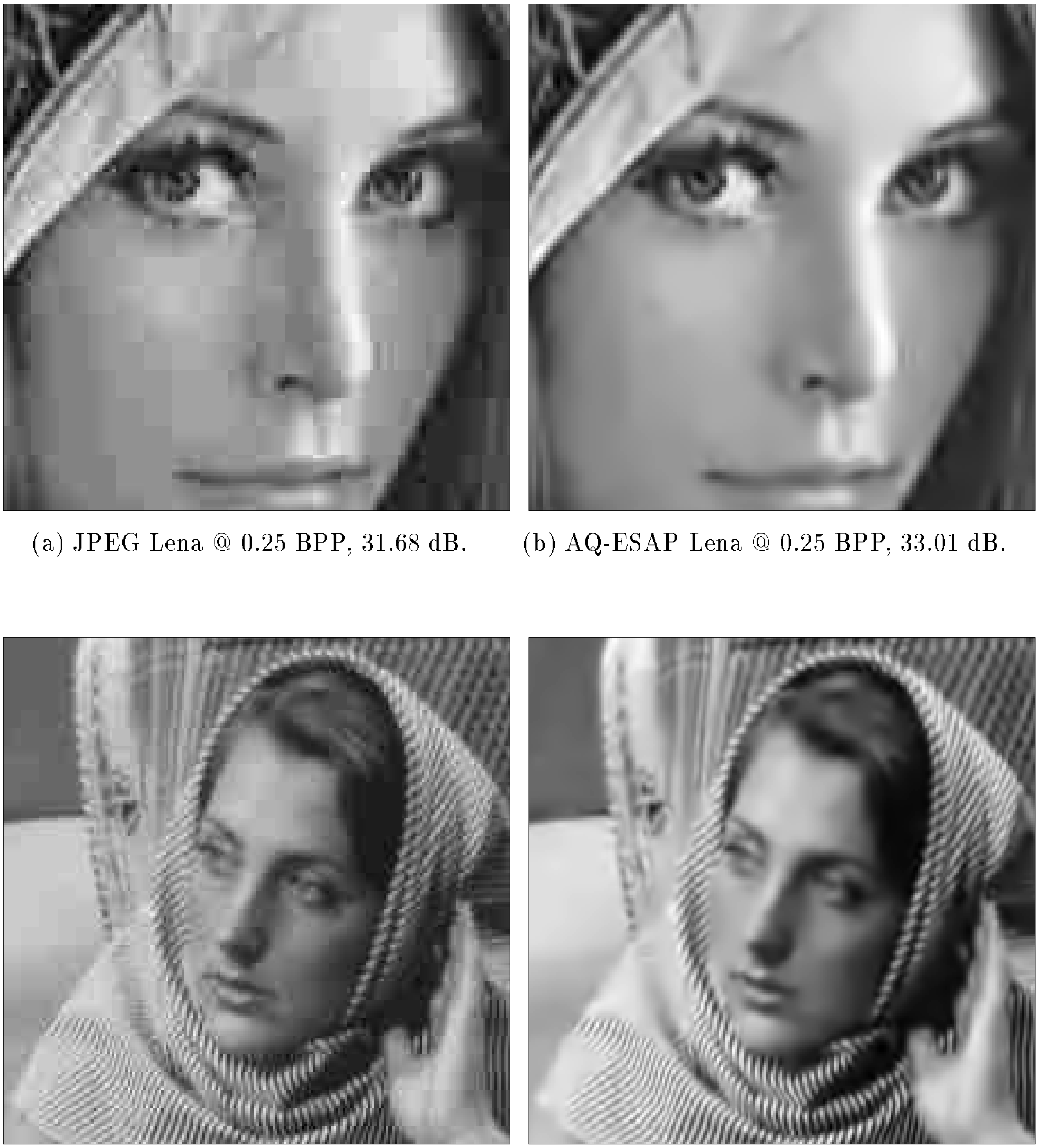
CHAPTER 4

Postfiltering Methods

This chapter presents several applications of the ESAP algorithm discussed in Chapter 3 to various block-based image coders. These coders include: monochrome and color JPEG, the lapped orthogonal transform (LOT), vector quantizers (VQ), and projections onto convex sets (POCS). For JPEG we consider both, default DCT coefficient quantization and image-adaptive DCT coefficient quantization. Image-adaptive quantization requires a preprocessing pass to optimize the rate-distortion product at the desired bit rate prior to image encoding. Additionally, in this chapter we explore synthetic aperture radar (SAR) image postfiltering with default and image-adaptive coefficient quantization. Within the SAR section, we also consider the effect of a weighted mean square error norm on the perceptual quality of the decoded image.

4.1 ESAP Enhancement of Baseline JPEG, AQ and JO Images.

In this section, we compare the objective performance of several versions of the JPEG-ESAP algorithm. We use two PSNR references: JPEG and the embedded zerotree wavelet (EZW) [41]. We also show the subjective improvement, including edge preservation with blocking reduction for the Lena and Barbara images. Figure 4.1(a) shows a baseline JPEG image of Lena at 0.25 BPP, 31.68 dB PSNR and Figure 4.1(b) shows its corresponding adaptive quantization (AQ-ESAP) image at 33.01 dB PSNR. Similarly, Figure 4.1(c) is a baseline JPEG Barbara image at 0.5 BPP, 28.27 dB PSNR and Figure 4.1(d) is its jointly optimized (JO-ESAP) version at 31.23 dB PSNR.



(a) JPEG Lena @ 0.25 BPP, 31.68 dB.

(b) AQ-ESAP Lena @ 0.25 BPP, 33.01 dB.

(c) JPEG Barbara @ 0.5 BPP, 28.27 dB.

(d) JO-ESAP Barbara @ 0.5 BPP, 31.23 dB.

Figure 4.1: Segments of the 512×512 JPEG, AQ-ESAP and JO-ESAP images.

BPP	JPEG	CBS	ESAP	AQ	AQ-ESAP	JO	JO-ESAP	EZW
0.25	31.68	31.64	32.76	31.88	33.01	32.34	33.07	33.17
0.50	34.90	34.87	35.59	35.48	36.23	35.96	36.45	36.28
1.00	37.96	37.95	38.37	38.88	39.23	39.58	39.61	39.55

Table 4.1: PSNRs for 512 × 512 Lena image.

BPP	Δ_{CBS}	$\Delta_{JPEG-ESAP}$	Δ_{AQ}	$\Delta_{AQ-ESAP}$	Δ_{JO}	$\Delta_{JO-ESAP}$	Δ_{EZW}
0.25	-0.04	1.08	0.20	1.33	0.66	1.39	1.49
0.50	-0.03	0.69	0.58	1.33	1.06	1.55	1.38
1.00	-0.01	0.41	0.92	1.27	1.62	1.65	1.59

Table 4.2: PSNR improvement over baseline JPEG for the 512 × 512 Lena image.

BPP	JPEG	CBS	ESAP	AQ	AQ-ESAP	JO	JO-ESAP	EZW
0.25	25.02	25.01	25.79	26.02	26.96	26.66	27.05	26.77
0.50	28.27	28.25	29.44	29.99	31.03	30.63	31.23	30.53
1.00	33.10	33.09	34.01	35.22	35.84	35.94	36.33	35.14

Table 4.3: PSNRs for 512 × 512 Barbara image.

BPP	Δ_{CBS}	$\Delta_{JPEG-ESAP}$	Δ_{AQ}	$\Delta_{AQ-ESAP}$	Δ_{JO}	$\Delta_{JO-ESAP}$	Δ_{EZW}
0.25	-0.01	0.77	1.00	1.94	1.64	2.03	1.75
0.50	-0.02	1.17	1.72	2.76	2.36	2.96	2.26
1.00	-0.01	0.91	2.12	2.74	2.84	3.23	2.04

Table 4.4: PSNR improvement over baseline JPEG for the 512 × 512 Barbara image.

Tables 4.1, 4.2, 4.3, and 4.4 summarize the decibel (dB) PSNR comparative results.

Let us take for example the 512×512 Lena image at 0.25 BPP. Referring to the first two lines of Tables 4.1 and 4.2, we observe that the JPEG's crossblock smoothing (CBS) reduces the PSNR by 0.04 dB. After applying ESAP to the default quantized JPEG image we obtain a 1.08 dB improvement. The postfiltering improvement when we *preprocess* the image to obtain an image-adaptive Q-table and *postprocess* it with ESAP is shown under the $\Delta_{AQ-ESAP}$. For this case it is 1.33 dB. To conclude this example, preprocessing the image to obtain a joint-optimized Q-table followed by ESAP postprocessing generates an improvement $\Delta_{JO-ESAP}$ of 1.39 dB. For comparison, the last column shows the EZW Δ_{EZW} improvement. For this case it is 1.49 dB.

4.2 ESAP and the Lapped Orthogonal Transform

In reference [64], we slightly modified ESAP to postprocess lapped orthogonal transform (LOT) images [54]. A comparison of Figures 4.2(a) and (b) shows that the main lobe's in-band frequency selectivity of the LOT basis functions is very similar to that of the DCT. Therefore, LOT images can be postfiltered exactly in the same manner the DCT decoded image is postfiltered, as explained in Section 3.1. After postfiltering a 256×256 LOT Lena image coded at 0.25 BPP and 0.5 BPP, we obtained approximately 0.9 dB PSNR gain for both cases. Two example images are listed in Appendix A.

4.3 Extension of ESAP to VQ Images

We also explicitly generated the DCT coefficients for the VQ images. As shown in Table 4.5, we obtained a 0.45 dB PSNR improvement over the VQ images after postfiltering. Please observe that the JPEG objective quality is better than VQ, even after ESAP postfiltering. Subjectively, the JPEG images are better than their ECRVQ-ESAP counterparts too. The results for 1.0 BPP were not computed because

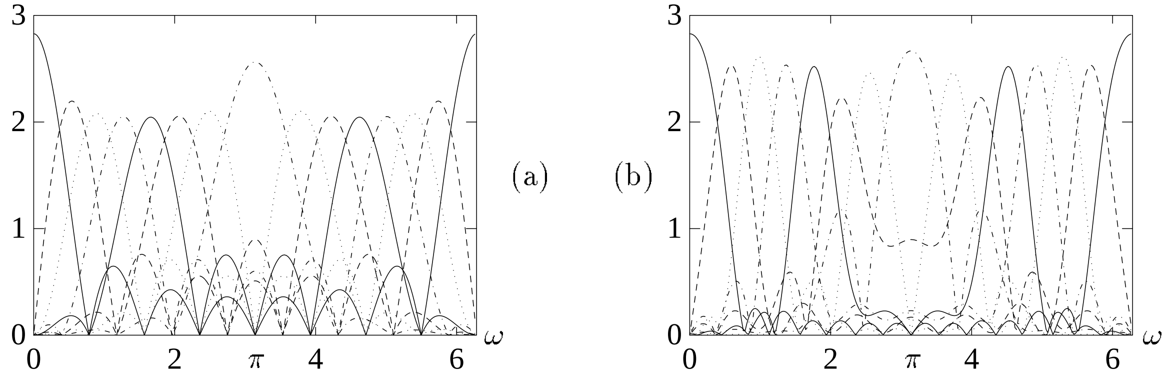


Figure 4.2: Magnitude of the Fourier transforms of the (a) DCT and (b) LOT.

the VQ codebook generation at that rate is impractical. Please refer to Appendix A for the example images corresponding to this section.

4.4 ESAP and POCS

In our implementation of POCS, we substituted the lowpass filter convex set S_1 with the ESAP constraint set. Furthermore, we set all the DCT quantization step sizes to “1” within the original bounds of the DCT quantization interval S_2 convex set. A rather interesting result was the *sharpening* of the edges, but only at the expense of a lower PSNR. For instance, Figure 4.3 shows the 0.25 BPP POCS-ESAP image after 13 iterations. The PSNR dropped to 29.66 dB, less than the original baseline JPEG's 31.68 dB at that bit rate. Nevertheless, the edges were sharper and there was no visible blocking. Going beyond the 13^{th} iteration over-emphasizes the edges and any perceptual improvement gain begins to be lost.

Figure 4.4 shows the flowchart of the POCS-ESAP algorithm. To start the POCS iterative process, the iteration i is set to 1. An initial ESAP postfiltered image x_1 is then JPEG compressed into $\hat{X}_i$, subsequently JPEG decompressed into $\hat{x}_i$ and then ESAP postfiltered to obtain $\tilde{x}_i$. ESAP constitutes our S_1 convex set. The algorithm then performs a convergence test between the current postfiltered image $\tilde{x}_i$ and the

BPP	JPEG	ECRVQ	ECRVQ-ESAP	Δ_{ECRVQ}	Δ_{ESAP}	Δ_{TOTAL}
0.25	31.68	29.28	29.72	-2.40	0.44	-1.96
0.50	34.90	33.44	33.91	-1.46	0.47	-0.99
1.00	37.96	N/A	N/A			

Table 4.5: PSNRs for 512×512 Lena ECRVQ image.

previous iteration postfiltered image $\tilde{x}_{i-1}$. If the MSE is less than some specified threshold δ, then STOP. Convergence has been attained. Otherwise, increment the iteration counter ($i = i + 1$), recompress the image with a unity Q-table (all 1's) to obtain $\tilde{X}_i$, and decompress the image also using a unity Q-table to obtain $\hat{\tilde{X}}_i$. Our convex set S_2 is the set of the original quantization bounds computed in the first iteration and it is denoted by $\hat{X}_1$ to the right of the sample-and-hold switch "sw". The $P_Q(\cdot)$ quantization projection block brings back into the initial quantization range any coefficient that falls outside the range. The resulting projected image transform $X_{i_{QC}}$ is inverse transformed (IDCT) and fed back into the top of the loop as a new x_i to begin the next iteration.

4.5 Application of ESAP to SBC — Explicit DCT Coefficient Generation

In this section we report on *explicit* DCT coefficient generation to postfilter SBC decoded images. In Chapter 6 we will discuss how the $\omega[\mathbf{m}]$ pixel-adaptive bandwidths are *implicitly* generated directly from the SBC coefficients.

The explicit coefficient generation method applies JPEG to synthesized subband coded images in order to explicitly generate the DCT coefficients necessary to drive ESAP's adaptive postfilter. Please note that unlike the DCT and the LOT, whose "pixel bandwidths" were directly estimated from the transform coefficients, we did not estimate the pixel bandwidths directly from the subbands in this section. After

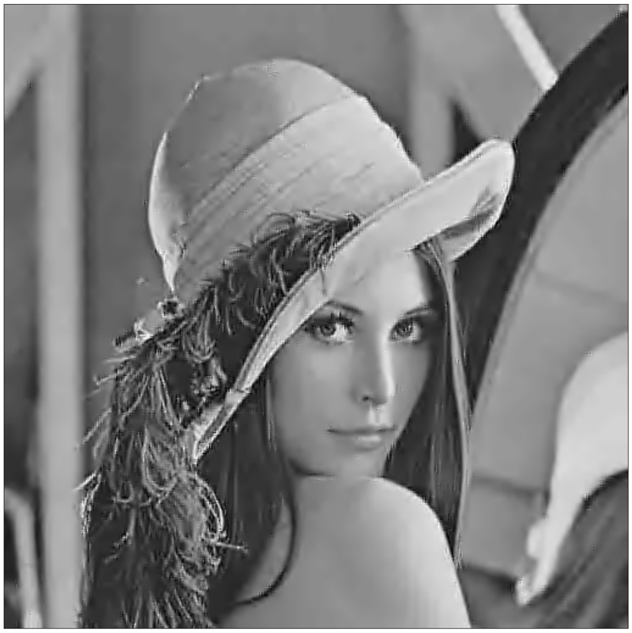


Figure 4.3: Resulting POCS-ESAP image at the 13^{th} iteration.

postprocessing, we obtained improvements of 0.14 and 0.11 dB PSNR over the 0.25 and 0.5 BPP SBC Lena images, respectively, as shown in Table 4.6. We could not improve the 1.0 BPP SBC PSNR, but we did not diminish its quality either. The 0.25 BPP image was coded with vector SBC CEC-RVQ while the 0.5 and 1.0 BPP images used scalar SBC CEC-RSQ (Section 2.6).

4.6 Enhancement of Image Sequences

The still-image postprocessing ideas presented in this thesis can be extended to MPEG image sequences. We carried out a simulation in which several 256 × 256 image frames extracted from the standard SIF testing sequence *Alexis* were intracoded (**I**) and then ESAP postprocessed. The compression ratio was about a 12:1 and the postfilter *steepness-bias* parameters [65] were $(S, B) = (5, 0)$. In [65] both interpolated frequency images are weighted with the log-sigmoid function

$$logsig(S, \omega, B) = \frac{1}{1 + e^{-[S(\omega - 1/2) + B]}} \tag{4.1}$$

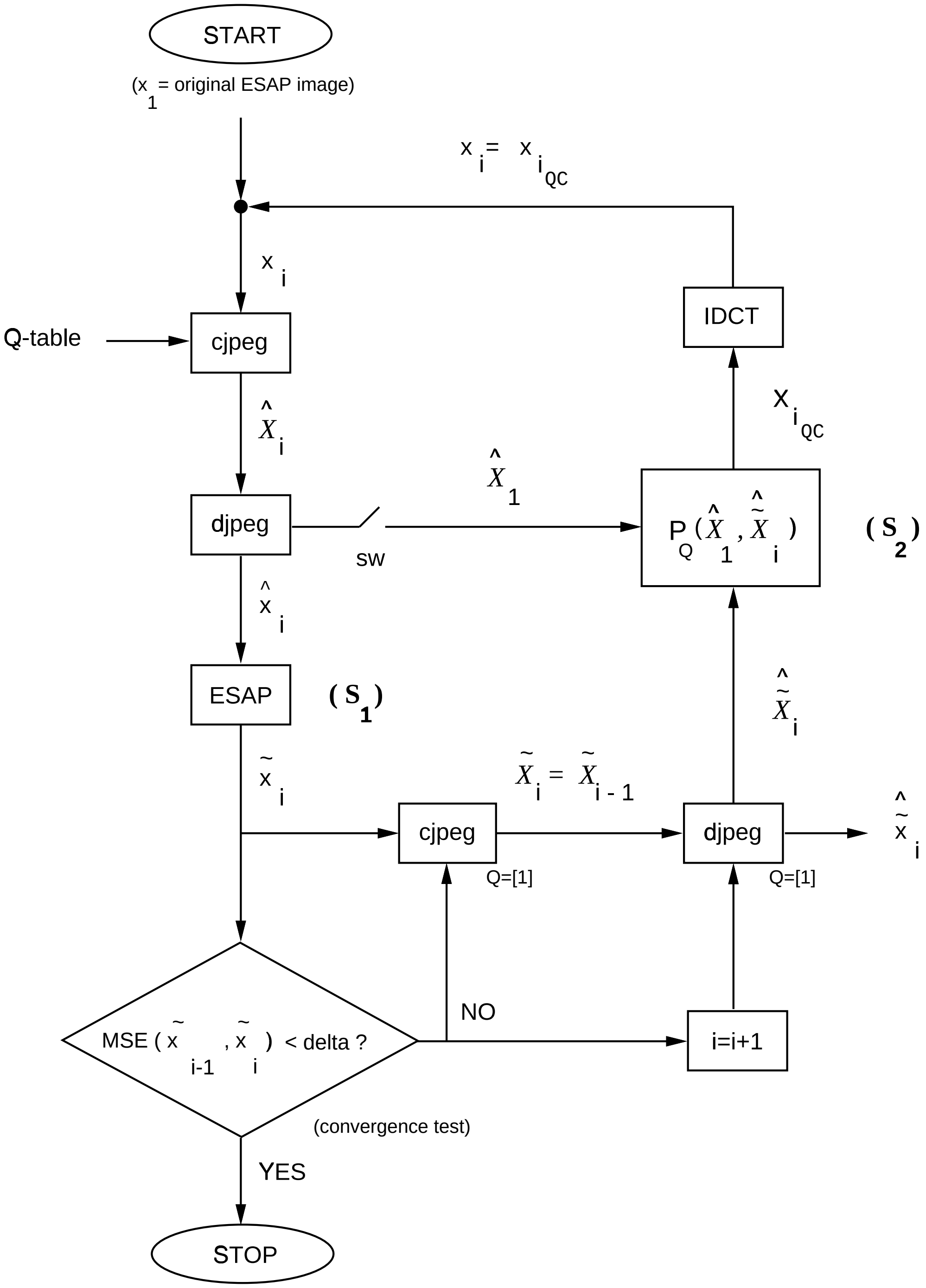


Figure 4.4: POCS-ESAP algorithm's flowchart.

BPP	JPEG	SBC CECRSQ/RVQ	SBC-ESAP	Δ_{SBC}	$\Delta_{SBC-ESAP}$	Δ_{ESAP}
0.25	31.68	34.10	34.24	2.42	2.56	0.14
0.50	34.90	36.89	37.00	1.99	2.10	0.11
1.00	37.96	40.13	40.13	2.17	2.17	0.00

Table 4.6: PSNRs for 512 × 512 Lena SBC image.

where S and B are the steepness and bias to be determined during encoding and ω is the normalized frequency [0..1]. The optimal (S, B) parameters are iteratively searched over the PSNR surface to obtain the maximum PSNR. The (S, B) values are then transmitted to the decoder as a byte of side information. The PSNR gain for the frame sequence averaged 0.4 dB PSNR. This was enough to significantly attenuate the mosquito effect visibly present in the original **I**-coded sequence.

4.7 Pixel-variable Region of Support for Adaptive Filter $h_{\mathbf{m},\mathbf{z}}[\mathbf{n}]$.

We also modified ESAP to obey a pixel-adaptive filter size $N_1 \times N_2$ similar to the Wavelet's theory constant-Q ($\frac{\Delta_f}{f}$ = constant) space-frequency localization property. We defined a new two-element space-frequency separable vector inner product

$$[\mathbf{w}]\ [\ \mathbf{f}[\mathbf{n}]\ ]^T = \begin{bmatrix} w_1[n_1] \\ w_2[n_2] \end{bmatrix} [\ f_1[n_1]\ f_2[n_2]\ ] = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \mathbf{Q}_{const} \qquad (4.2)$$

where $1 \leq w_1 \leq N_{1_{max}}$, $1 \leq w_2 \leq N_{2_{max}}$ and the DCT interpolated bandwidths $f_1[n_1]$ and $f_2[n_2]$ are in the (0.0, 1.0) frequency range. This variant of the algorithm did not transmit any $\mathbf{w}[\mathbf{n}]$ side information since $\mathbf{f}[\mathbf{n}]$ is implicitly embedded in the DCT coefficients.

Initially, this modification seemed to be supported by previous tests, where the optimally searched scalar w parameter of the EDGE pixels varied between 7 to

13 pixels most of the time, never reaching its maximum width of 17. Also, many TEXTURE pixels with high bandwidths were left unchanged ($f = 1.0$), strongly suggesting a constant-Q hypothesis.

After various experiments, we realized that the pixel-variable region of support (ROS) of Equation (4.2) was not performing better than the fixed 17×17-pixel region of support. We attribute this behavior to the artificially imposed 8×8 MCU segmentation. A 17×17 adaptive filter will always cover at least four contiguous MCUs, smoothing out any artificial discontinuities between the four blocks. On the other hand, smaller adaptive filters of order 3×5 for example, may not be able to smooth out interblock discontinuities.

In an experiment with the 512×512 JO Lena image coded at a bit rate of 0,25 BPP, we found that convergence was attained for a pixel-variable ROS with parameters[1] $(\sigma, t, f, Q) = (1.7, 1, 0.25, 1.7)$. The resulting postfiltered PSNR was 32.86 dB, which is 0.21 dB lower than the fixed 17×17 ROS JO-ESAP result of 33.07 dB shown in Table 4.1. The resulting image is listed in Appendix A under Section 4.7.

4.8 JPEG Color Image Enhancement

An obvious extension of the gray-scale ESAP algorithm is postfiltering in the RGB or YC_bC_r color spaces. We perform a straightforward separate postfiltering optimization of each color plane.

For this experiment, the maximum color image size is limited to 1008×1008 pixels. This restriction arises from the Canny edge detection software which limits the image size to 1024×1024 pixels for proper handling of the DC values and boundary conditions. Since we need to reserve an eight-pixel stripe on each side of the image to accommodate the reflected borders (one JPEG block on each side), the effective

[1] Please refer to Section 3.4 for parameter definitions.

maximum width for this simulation is limited to 1008 × 1008. The entire Canny software was not recompiled to allow larger images since we felt that 1008×1008 pixels was more than adequate for demonstration purposes. Moreover, the computational load increases with an $O(N^2)$, which imposes extremely long processing time for large images ($N > 512$). Finally, the images are required to be multiples of (16 × 16)-pixel JPEG minimum coded units. This is a restriction imposed by the CCIR-601 YUV-3 color space sampling of 2:1:1.

In order to manually create images that are 16×16 MCU multiples we developed two additional auxiliary functions, `16x16_MCU_pad.c` and `cut_image.c` with usage:

```
16x16_MCU_pad  Hin Win Hout Wout infile outfile
cut_image      Hin Win Hout Wout start_row start_col infile outfile
```

The source code for the above functions is available in Appendix B. We also made extensive use of the *ImageMagick Toolkit* ©1994 by E. I. duPont de Nemours & Company. This toolkit provides very useful image format conversion utilities such as *convert* and *display*. To process an RGB color image it is necessary to perform the steps listed in the file `PROCEDURE_color_esap` included in the same appendix. This procedure requires of the following steps:

1. Convert RGB image to PNM.

2. Compress PNM image using `cjpeg_5b`

3. Decompress image with modified `djpeg_5b_esap_color` program to obtain the Y luminance `DCT_COEFFS.0` dump and the U and V chrominance `DCT_COEFFS.1` and `DCT_COEFFS.2` PNM coefficient dump.

4. Convert PNM image to YUV-3 image

5. If necessary, manually coerce any resulting gray-scale Y, U, or V images to be multiples of 16×16 macroblocks using the utilities `cut_image` or `16x16_MCU_pad`.

6. Convert the original PNM image to PGM (gray-scale). This is required since the experimental program `djpeg_5b_esap_color` dumps the Y coefficients in MCU order (1–2,5–6,9–10,13–14,...) and not in the expected (1,2,3,4,...) sequential order. Save the output `DCT-COEFFS.0` file.
7. Postfilter each color plane using `esap_r29`.
8. Compute each color component MSE using Equation 4.3.
9. Obtain composite MSE using Equation 4.4.
10. Obtain CPSNR using Equation 4.5.
11. For a subjective evaluation, display the original, JPEG decoded and ESAP postfiltered YUV-3 images.

4.8.1 Composite PSNR Distortion Measure

To assess the objective quality of JPEG color images we need to define a composite MSE (CMSE) and its corresponding composite PSNR (CPSNR). Recalling that the 2:1:1 YC_bC_r or equivalent YUV-3 color space subsampling spatially decimates the C_b and C_r color planes by a factor of two, we observe that the Y luminance component contributes 4/6 to the composite MSE while the C_b and the C_r chrominance components each contribute 1/6 of the weight. The MSE for each color component of the original $x[\mathbf{n}]$ and decoded $\hat{x}[\mathbf{n}]$ images under consideration is given by

$$d_{MSE}(x_{comp}[n_1,n_2],\hat{x}_{comp}[n_1,n_2]) = \frac{1}{N_1N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}(x_{comp}[n_1,n_2]-\hat{x}_{comp}[n_1,n_2])^2. \tag{4.3}$$

The composite MSE distortion is thus defined by

$$\begin{aligned} d_{CMSE}(x_{YC_bC_r}[\mathbf{n}],\ \hat{x}_{YC_bC_r}[\mathbf{n}]) &= \frac{2}{3}\, d_{MSE}(x_Y[\mathbf{n}],\hat{x}_Y[\mathbf{n}]) \\ &+ \frac{1}{6}\, d_{MSE}(x_{C_b}[\mathbf{n}],\hat{x}_{C_b}[\mathbf{n}]) \qquad (4.4) \\ &+ \frac{1}{6}\, d_{MSE}(x_{C_r}[\mathbf{n}],\hat{x}_{C_r}[\mathbf{n}]) \end{aligned}$$

BPP	**Original**	**JPEG**	**ESAP**	Δ	**Iterations**
	Image	**CPSNR** (dB)	**CPSNR** (dB)	(dB)	(Y, U, V)
0.21	lena.pnm	lena.21.pnm	lena.21.esap.pnm	+0.397	63,35,7
		30.905	31.302		
0.30	lena.pnm	lena.3.pnm	lena.3.esap.pnm	+0.672	91,21,133
		31.765	32.437		

Table 4.7: CPSNR results for 512×512 24-BPP LENA RGB image.

where the 2-D coordinate $[\mathbf{n}] = [n_1, n_2]$. Now the CPSNR can be expressed as

$$\mathrm{CPSNR} = 10 \log_{10} \left(\frac{255^2}{d_{CMSE}(x_{YC_bC_r}[\mathbf{n}], \hat{x}_{YC_bC_r}[\mathbf{n}])} \right). \tag{4.5}$$

Using the above CPSNR definition, the default JPEG Q-tables, and the 512×512 *RGB* Lena image we obtained the results shown in Table 4.7. The location of these *.pnm* images is listed in Appendix A.

4.9 Application of ESAP to SAR Images

This section presents the results obtained by precomputing image-adaptive Q-tables for one of the X-band images of the MSTAR 16-BPP Public Clutter SAR Set [71] compressed at 4, 2, 1, 0.5 and 0.25 BPP and by postfiltering these images with ESAP afterwards. This is another case of image preprocessing followed by adaptive postprocessing. The spatial resolution of this image is 30 cm. The selected image is:

Image	**hb06194**
Header length	01569
Number of columns	1478
Number of rows	1784
Site	huntsville_al
Desired depression	15 deg
Size	10549089 bytes
Resolution	16 BPP

To view this image please refer to Appendix A.

One of the images with the most varied features in the set is **hb06194**. It contains several man-made structures rich in high-frequency detail. These include about eight buildings and houses, fences, driveways, streets and roads. Noticeably, about 25% of the image seems like highly textured plowed land. These details, which might as well be sown ground, are displayed in a diagonal spatial orientation. The image also contains about 30% of low frequency regions corresponding to wooded areas. The image dynamic range is first converted from 16 to 8-BPP using the `16to8bpp()` program. Within this program, we took the $\log_{10}()$ of the image to further reduce its dynamic range and make the details more visible. In its original linear scale, only a few features are discernible, that is, the image mostly looks like dark background. Finally, we cropped the image to 1008×1008 pixels in order to make it compatible with the ESAP Canny edge detection algorithm.

SAR-AQ-ESAP Results

Table 4.8 shows the SAR-AQ-ESAP results. Letter q refers to the JPEG quality and the value λ optimizes the Langrangian cost function $J(\lambda) = D + \lambda R$. Blank entries are not applicable (N/A). The location of the *.pgm* images in the second column of Table 4.8 can be found in Appendix A.

We can draw some conclusions by comparing default Q-table JPEG compressed SAR images with the same images coded with image-adaptive Q-tables (preprocessing) and subsequently, postfiltered with ESAP (postprocessing). SAR-AQ works well *numerically* by significantly improving the PSNR. However, subjectively, JPEG and AQ images have about the same perceptual quality. Adding the extra step of postfiltering a SAR-AQ image with ESAP does not improve the *perceptual* SAR image quality either. Our findings for this scenario are:

1. For our $\log_{10}$ SAR image coded at bit-rates of 4, 2 and 1 BPP the corresponding AQ PSNR increases are 5.6, 2.5 and 0.5 dB over JPEG, respectively. The ESAP improvement is negligible in all cases. Subjectively, the visual quality differences

Method	**Image**	**BPP**	**PSNR**	Δ_{AQ}	Δ_{ESAP}
	(.pgm)		(dB)	(dB)	(dB)
JPEG (q=92)	s4j	3.982679	38.307110		
AQ (λ=725)	s4a	3.971940	43.940041	+5.632931	
ESAP	s4e	3.971940	43.940044		+0.000003
JPEG (q=71)	s2j	2.000476	29.326717		
AQ (λ=11929)	s2a	2.000606	31.868393	+2.541676	
ESAP	s2e	2.000606	31.868401		+0.000008
JPEG (q=35)	s1j	0.993944	26.482403		
AQ (λ=25000)	s1a	0.993605	26.999495	+0.517092	
ESAP	s1e	0.993605	26.999506		+0.000011
JPEG (q=16)	s.5j	0.489941	24.839300		
AQ (λ=43212)	s.5a	0.487311	24.906841	+0.067541	
ESAP	s.5e	0.487311	25.028065		+0.121224
JPEG (q=9)	s.25j	0.236983	23.588264		
AQ (λ=73993)	s.25a	0.233314	23.606958	+0.018694	
ESAP	s.25e	0.233314	23.715248		+0.10829

Table 4.8: PSNR comparisons for the hb06194 SAR image using JPEG, AQ and ESAP.

among the three cases—JPEG, AQ and ESAP—are almost imperceptible.

2. For the 0.5 bit-rate, the visual quality of the JPEG image is perceptually better (sharper) than the AQ and ESAP images although its PSNR is 0.19 dB lower than the ESAP image. DCT coefficient quantization visibility may explain this mismatch.

3. For 0.25 BPP, all the images have very low perceptual quality, although AQ and ESAP showed a 0.02 and 0.12 dB PSNR improvement over JPEG, respectively.

4.9.1 Weighted MSE

We also experimented with weighting of the DCT coefficient errors using the tool `adaptQw`. This program implements a weighted mean square error (WMSE) criterion in the DCT domain by assigning different weights to the errors corresponding to each 2-D DCT frequency. The interested reader may find the modified `adaptQw.c` and `qadaptw.c` programs that perform WMSE image-adaptive Q-table computation in Appendix B.

The next four Qweight[n] matrices are actual examples applied to a 128×128 subimage of the larger SAR **hb06194** image. All examples are coded at about 0.5 BPP. For a given bit rate, larger coefficient weighting caused proportionally larger DCT step sizes as shown in the example pairs

$$\text{Qweight[n]} \rightarrow \text{Qtable[n]}$$

below. The first Qweight[0] weighting matrix is the MMSE case. An explanation of this behavior is offered after the examples.

Examples of Qweight[n] Matrices

Qweight[0]

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Qtable[0]

PSNR 24.503012 BPP 0.482221

29	37	51	41	95	255	255	255
42	50	55	77	118	255	255	255
46	56	72	94	255	255	255	255
51	58	110	255	255	255	255	255
46	102	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Qweight[1]

1	1	1	1	1	1	1	1
1	1	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2

Qtable[1]

PSNR 24.274549 BPP 0.494398

29	33	36	36	65	255	91	255
26	34	92	255	255	255	255	255
30	112	255	255	255	255	255	255
51	255	255	255	255	255	255	255
46	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
88	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Qweight[2]

1	1	1	1	1	1	1	.5
1	1	1	1	1	1	.5	.5
1	1	1	1	1	.5	.5	.5
1	1	1	1	.5	.5	.5	.5
1	1	1	.5	.5	.5	.5	.5
1	1	.5	.5	.5	.5	.5	.5
1	.5	.5	.5	.5	.5	.5	.5
.5	.5	.5	.5	.5	.5	.5	.5

Qtable[2]

PSNR 24.546888 BPP 0.662445

46	49	61	70	95	255	90	45
62	71	77	105	66	255	32	255
55	69	72	94	255	31	57	62
71	77	110	255	30	52	255	255
72	87	255	35	51	77	255	255
255	255	32	65	69	255	255	255
255	33	43	75	255	255	255	255
39	46	255	255	255	255	67	255

Qweight[3]

1.5	1.0	0.9	1.5	1.5	3.6	4.6	5.5
1.1	1.1	1.3	1.7	2.4	5.3	5.5	5.0
1.3	1.2	1.5	1.5	3.6	5.2	6.3	5.1
1.3	1.5	2.0	2.6	4.6	7.9	7.3	5.6
1.6	2.0	2.5	5.1	6.2	9.9	9.4	7.0
1.5	3.2	5.0	5.8	7.4	9.5	10.3	8.4
4.5	5.8	7.1	7.9	9.4	11.0	10.9	9.2
6.5	8.4	8.6	8.9	10.2	9.1	9.6	9.0

Qtable[3]

PSNR 24.385801 BPP 0.483682

37	33	33	55	95	255	255	255
26	34	53	105	255	255	255	255
30	56	90	94	255	255	255	255
51	80	255	255	255	255	255	255
112	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Effects of the WMSE on the DCT Quantization Matrix

This behavior can be explained as follows. Recall that the WMSE is a summation of the weighted square errors of the quantized DCT coefficients. Let us assume that we change the weight of a single coefficient while keeping the remaining 63 coefficient errors with unity weight. Since we are trying to minimize a summation, any coefficient error that is artificially increased by a larger weight, will tend to be offset by lower errors in the remaining 63 coefficients. The only way to achieve this is by *decreasing* the step sizes of the remaining 63 coefficients. Therefore, increasing the weight of any error associated with a particular coefficient has the effect of reducing the step sizes, and consequently the square errors, of the remaining coefficients in order to minimize the MSE at the desired bit rate. This effect is also true for groups of coefficients taken at a time (case of Qweight[1]). We also found that decreasing the weight of any coefficient error or group of coefficient errors (case of Qweight[2]) *increases* the step sizes of the remaining coefficients with respect to the MMSE case represented by the unity weights of Qweight[0]. The larger step sizes of the remaining coefficients cause a "larger" error to offset the artificially diminished importance of the small weight errors.

Our results did not show any obvious visual improvement over simple MMSE (equally-weighted error). It was difficult to reach a conclusion with the limited experimentation performed. It was not possible to identify a good Qweight matrix that would have improved the visual quality of the small 128×128 SAR sub-image under testing. Larger image sizes in the range of 512×512 to 1024×1024 pixels must be considered. We must also take into consideration other classes of images including natural images, head and shoulders scenes, and visible spectrum or infrared remote sensing images. This topic remains largely as an open research area.

CHAPTER 5

Pre-post Filtering Methods

In this chapter we extend the concepts of one-dimensional pre-post filtering of Section 2.2 to two-dimensional gray-scale images. The underlying idea of audio pre-post filtering is to raise the low level, high frequency components above the analog recording medium's high frequency hiss noise and then during playback, restore those components to their original level. By attenuating these high frequencies with a gain that is the reciprocal of the preemphasis gain, this process simultaneously brings down the preemphasized signal along with the noise level by an amount equal to the attenuation.

Similarly, the fundamental idea behind the *iterative pre-post filter* (IPF) to be discussed in Sections 5.2 and 5.3, is to emphasize the high frequencies with respect to the low frequencies before coding and then, deemphasize these frequencies after decoding. It is well known that perceptually the HVS tolerates — and sometimes does not even perceive — low bit rate blocking in the presence of high frequency spatial sinusoids with magnitudes above certain perceptual thresholds for each DCT frequency band [23, 24]. We should search for a preemphasis gain and highpass cutoff frequency that will effectively *raise* these frequency components above the blocking noise. Upon deemphasis, blocking should have been deemphasized by the preemphasis gain. By doing this we are preserving the high frequency, low level detail that will improve the visual quality as well as the PSNR.

This process imposes a tradeoff between the amplitude of the low frequency and the high frequency coefficients of the DCT (or a SBC in the next chapter). Since the overall dynamic range of the preemphasized image is compressed to an 8 BPP unsigned integer range, the net effect is a proportional decrease in the magnitude of the DC and near-DC coefficients while the relative magnitudes of the remainder AC coefficients are increased. This causes a larger proportion of the AC coefficients to become significant (non-zero) after quantization, improving the decoded image quality near the image edges and textures. The process is reversible at the decoder due to pre-post filters. The relation between the frequency responses of the two filters is given by

$$P(\omega, G)\, D(\omega, G) = 1. \tag{5.1}$$

where G is the preemphasis gain expressed in dB. The extension of this 1-D equation to images will be fully described in Section 5.3 through Equations 5.4–5.6.

This thesis is mainly concerned with improving the JPEG algorithm's performance in a coder-compliant manner. In other words, we look for ways to minimize the DCT's rate-distortion product. In JPEG image coding, as with almost any other image coder, the distortion is usually measured by the MSE or the PSNR since both are easily tractable mathematical expressions. At low bit rates the blocking noise is caused by the coarse quantization of all the DCT coefficients. The blocking discontinuities are more noticeable in the low frequency regions with spatial frequencies around 8 cycles/degree (Section 2.4.2). Since ESAP does a reasonable job in postfiltering the coarsely quantized DC and low-frequency AC coefficients in the background, we have that portion of the problem solved. We now need to raise the level of the high frequency AC coefficients above their respective quantization thresholds in order to reduce the DCT blocking noise level and retain more of the high frequency content of the image.

We can interpret the blocking noise level as windowing (Section 3.1) or in the spatial domain as the absence of the high frequency components of the original signal. When we raise the level of those high frequency AC components sufficiently, so that the DCT quantization steps are relatively small enough in relation to the preemphasized signal, these coefficients become significant (non-zero) and help to diminish the artificial blocking discontinuities caused by independent MCU encoding.

We can increase the high frequency content of an encoded image in two ways. The first obvious one is to diminish the quantization step sizes of the high-frequency DCT coefficients. This introduces a tradeoff. With the first option, the bit rate will go up, therefore we need to increase the DC and low-frequency AC quantization step sizes to keep a constant bit rate. Now this causes further MSE distortion, therefore this obvious approach does not work.

The second approach involves pre-post filtering, dynamic range companding, adaptive quantization and ESAP postfiltering based on the dbx audio noise reduction system. This is the subject of Sections 5.1 and 5.2 below. With these approaches we are able to artificially raise the value of the AC coefficients before encoding forcing them to be significant. Since we are using pre-post filters, the image can be reversibly deemphasized at the decoder. The dynamic range companding brings the preemphasized dynamic range back to an 8 BPP range, and along with it, lower amplitude DC and low frequency AC coefficients. This is a reasonable tradeoff for artificially larger high frequency AC coefficients which preserve some of the detail which would be otherwise lost. The adaptive quantization of the DCT coefficients through Lagrange multiplier cost minimization [31, 32], adjusts the quantization table to accommodate larger AC coefficients at the expense of smaller DC and low frequency AC coefficients. Finally, the insertion of ESAP in one of three possible positions smoothes out the interblock discontinuities by interpolating the DCT bandwidths among at least four physically-contiguous 8×8 blocks.

This combination of techniques improves the JPEG objective and subjective quality to comparable state-of-the-art subband coders. This increase in quality, which preserves full JPEG compatibility, comes at a price though. The penalty paid is additional computation for the pre-post filtering, the image-adaptive Q-table computation, for dynamic range companding and for the ESAP adaptive convolution. Since IPF is an asymmetrical process, it iteratively searches for the optimal MSE parameters at the encoder. The search process, which follows a steepest descent path on the

MSE surface, will iterate until MMSE convergence. However, at the decoder, all this processing is done only once.

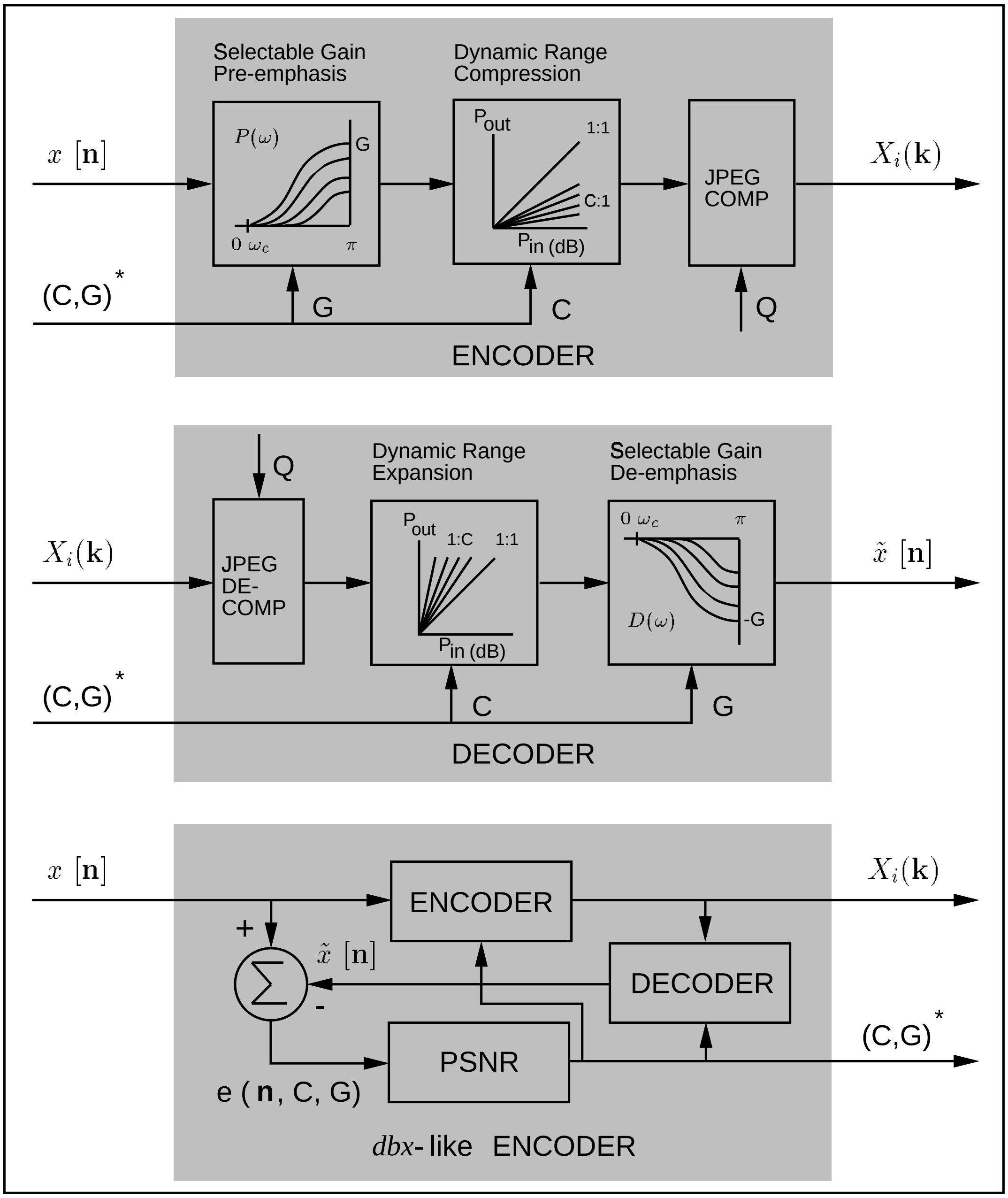
We feel that the proposed algorithms could be implemented in faster dedicated firmware to obtain real-time optimal performance. Suboptimal fast encoder performance can also be achieved with fixed pre-post filters and fast Q-table lookup based on classes of images and image statistics. Image companding (Equations 5.7 to 5.9) is a very fast $O(N^2)$ process that should not affect IPF's real-time performance significantly.

5.1 Initial *dbx*-like Pre-post Filtering System

Figure 5.1 shows an early version of the iterative *dbx*-like pre-post filtering algorithm. It was implemented in MATLAB using a combination of "m" functions and system calls. For example, using a cutoff frequency $\omega_c = \frac{3}{16}\pi$, we were able to obtain a 1.1 dB PSNR improvement over JPEG at 0.3 BPP. The optimal compression-gain $(C, G)^*$ was (-0.0625, 0.7500) dB. These "dbx" parameters actually *expanded* the dynamic range from $[0 \ldots 255]$ to $[0 \ldots 276]$ and *preemphasized* the image by $+0.75$ dB ($\times 1.19$ factor) before JPEG encoding. Please note that an expanded upper limit of the dynamic range does not necessarily imply that $x_{max} > 255$. At the decoder, the image's dynamic range was compressed back to its nominal $[0 \ldots 255]$ original range and subsequently deemphasized by -0.75 dB.

5.2 Iterative Pre-post Filtering of JPEG Images

The *dbx*-like test system of Section 5.1 was not completely adaptive in the sense that its quantization table remained fixed for all iterations. Only its pre- and postfilter gain, and its dynamic range companding were image-adaptive. In order to better conform to specific image statistics which change with each iteration's preemphasis values, the filter's highpass cutoff frequency ω_c and quantization table Q must be



NOTE: 2-D filters are separable 1-D $P(\omega)$ and $D(\omega)$.

Figure 5.1: Iterative image-adaptive *dbx*-like JPEG blocking noise reduction.

jointly optimized in conjunction with the pre-post filtering and companding. Also blocking reduction ESAP postfiltering can be inserted after JPEG decoding. The system would be asymmetric and require a minimal overhead on the order of 0.0001–0.0002 BPP. The system can also have a fast implementation by using FFTs for preemphasis and deemphasis and Fast DCTs for the JPEG codecs (already part of the IJG software.)

Using the basic Q-table adaptation algorithm provided by Crouse and Ramchandran [31] which was extended as described below, we developed the JPEG-compatible iterative pre-post filter with ESAP enhancement shown in Figure 5.2. The IPF algorithm is based on the *dbx* audio noise reduction system. The usage of the `adaptQw()` program is as follows:

```
> adaptQw
USAGE: adaptQw b|i|r bpp:in | lambda:in [lambda2:in lambda_inc] image:in
               height:in width:in Qlower:in Qupper:in Qweight:in Q:out

       Options: b = target bit-rate, [bpp] range (0.1..7.99)
                i = interactive, requires only lambda:in
                r = range, also requires lambda2:in lambda_inc

       Examples:
          adaptQw b 0.50 lena.512 512 512 Q.1 Q.255 Qw Q_0.5bpp
          adaptQw i 15000 lena.512 512 512 Q.26 Q.225 Qw Q_15000
          adaptQw r 15000 30000 1000 lena.512 512 512 Q.1 Q.255 Qw Q_15-30k

       This program minimizes the Lagrangian cost function
           J(lambda) =  D + lambda*R
       Higher lambda means lower rate.
```

Given a desired bit rate, the `adaptQw()` algorithm performs a binary `bpp` search between two initial points (λ_{min}, b_{max}) and (λ_{max}, b_{min}) obtained from the *empirical* exponential curve fit

$$\lambda = 10^{\,(8-b)/1.63} \tag{5.2}$$

or equivalently

$$b = 8 - 1.63 \log_{10}(\lambda) \tag{5.3}$$

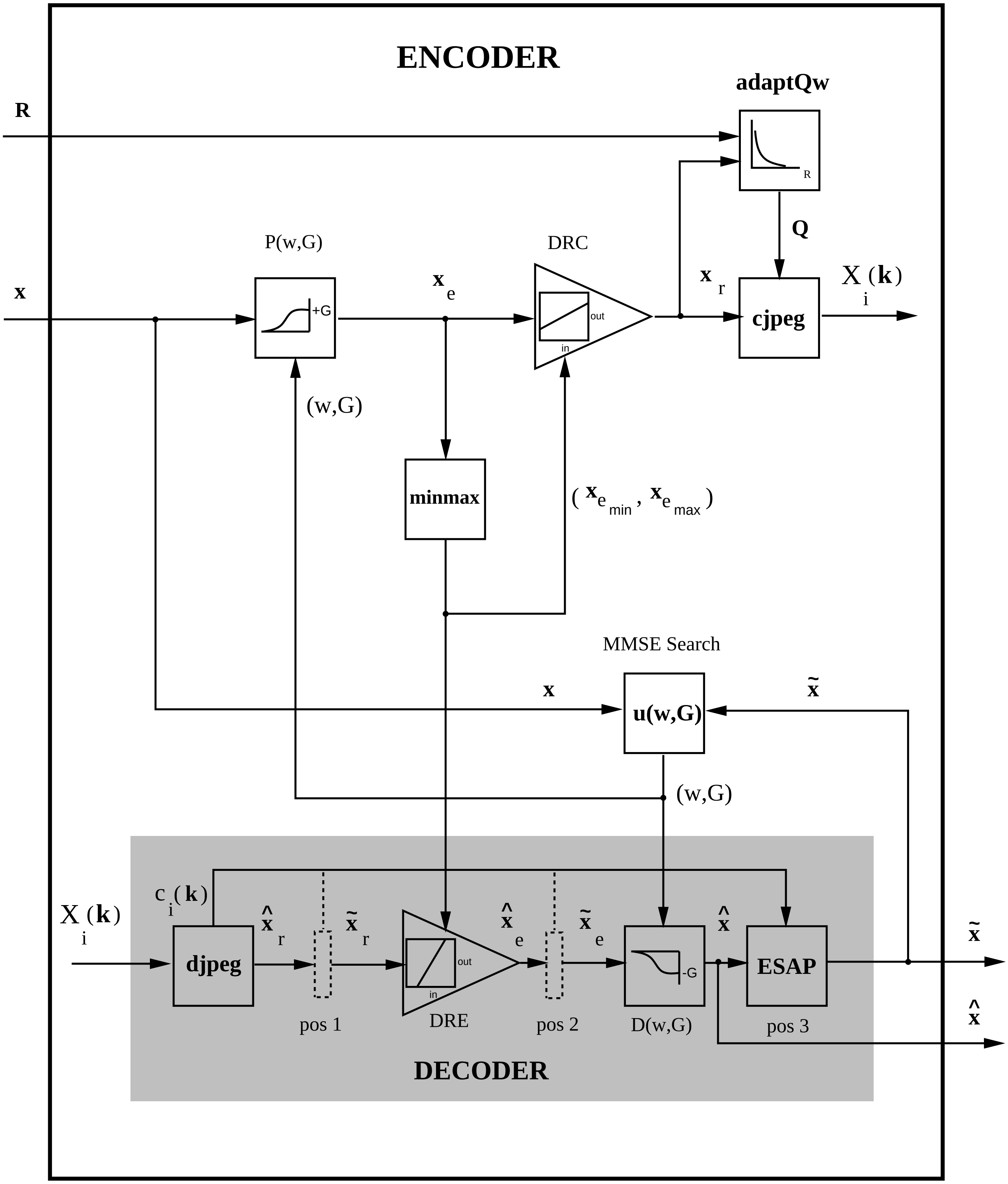


Figure 5.2: IPF JPEG blocking noise reduction.

where b represents the desired bpp rate. The program initializes λ_{min} to λ/s_f and λ_{max} to $\lambda \times s_f$, where s_f is an appropriate *scale factor* in the range $\{2, 4, 8, \ldots\}$. Other output options include interactive and range optimal Q-tables for any given λ.

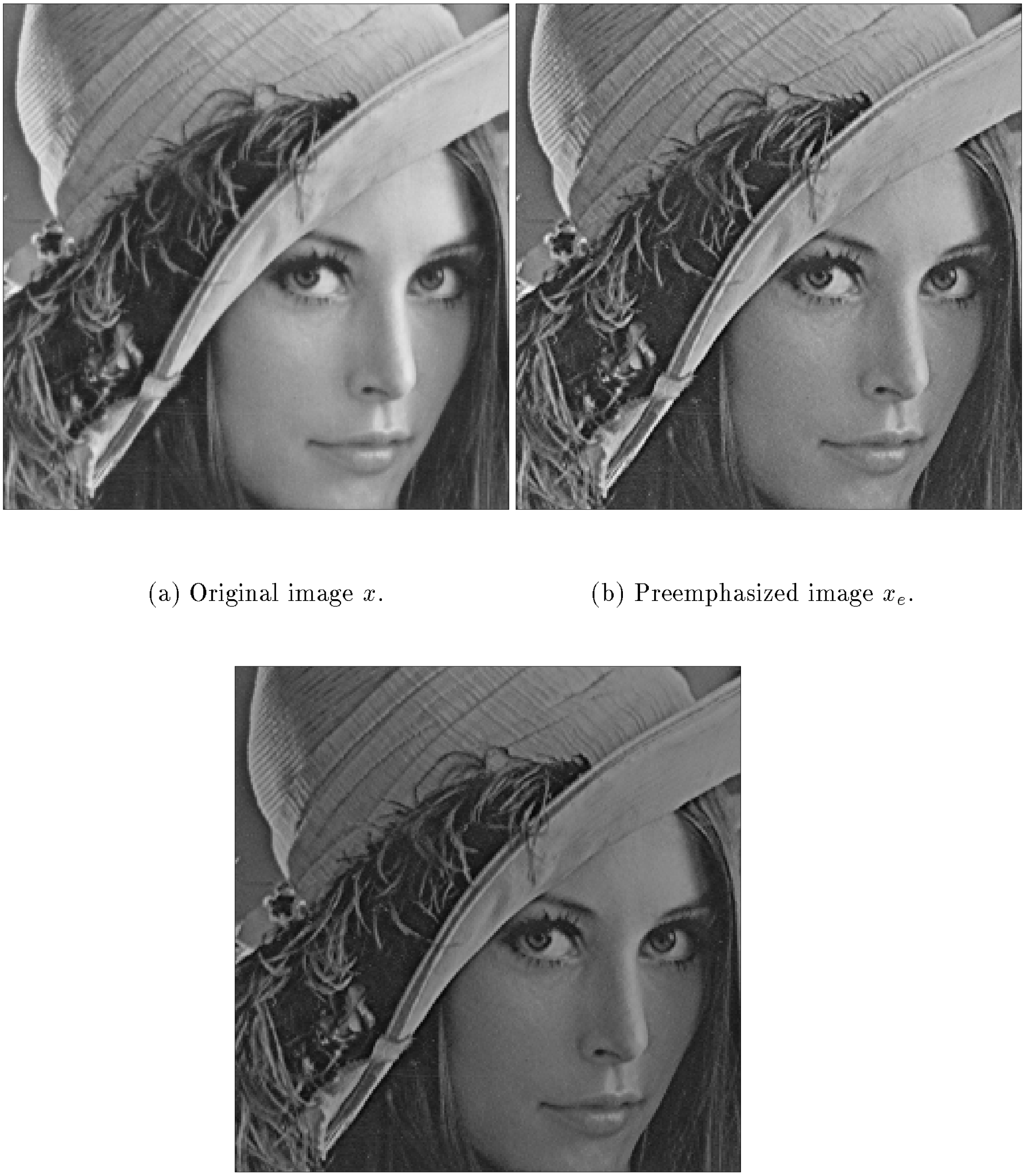
The initial version of the algorithm was completely implemented in MATLAB with the exception of system calls to cjpeg, djpeg, adaptQw, etc. It was later translated to C to prevent "MATLAB out of memory" allocation errors. The algorithm's source files in the C and MATLAB languages are listed in Appendix B.

5.3 IPF Description

IPF is best described by referencing Figures 5.2, 5.3, and 5.4. The image examples of Figures 5.3 and 5.4 correspond to the 0.25 BPP row of Table 5.1. It should be mentioned that the displayed dynamic range of Figures 5.3(b)–(c) and 5.4(d)–(e) is *simulated*. We used 12 bits to represent a signed integer range of $-2048 \ldots + 2047$ that quantizes the IPF's internal 32-bit floating-point precision. We kept the 8-BPP unsigned integer dynamic range of (a), (f), and (g) unchanged and rescaled the 12-BPP signed integer range of (b), (c), (d), and (e) into an 8-BPP unsigned range for proper image display when using a linear scale.

At the *encoder*, the original image x of Figure 5.3(a) is preemphasized with the filter $P(\omega, G)$, where ω is the HPF cutoff frequency corresponding to the separable DCT bandwidth f of Figure 3.1 ($\omega \approx f$) and G is the preemphasis gain in dB. The dynamic range of the resulting preemphasized signal x_e shown in Figure 5.3(b) is compressed into an 8-BPP range of 0-255 using the *dynamic range compression* (DRC) function. Then the range-compressed signal x_r of Figure 5.3(c) is used to compute a MMSE optimally quantized Q-table using the Lagrange multiplier minimization function `adaptQw()`. The image is then JPEG compressed using the standard IJG `cjpeg()` function.

The 2-D pre-post filters $P(\omega, G)$ and $P(\omega, G)$ are made of separable 1-D filters,

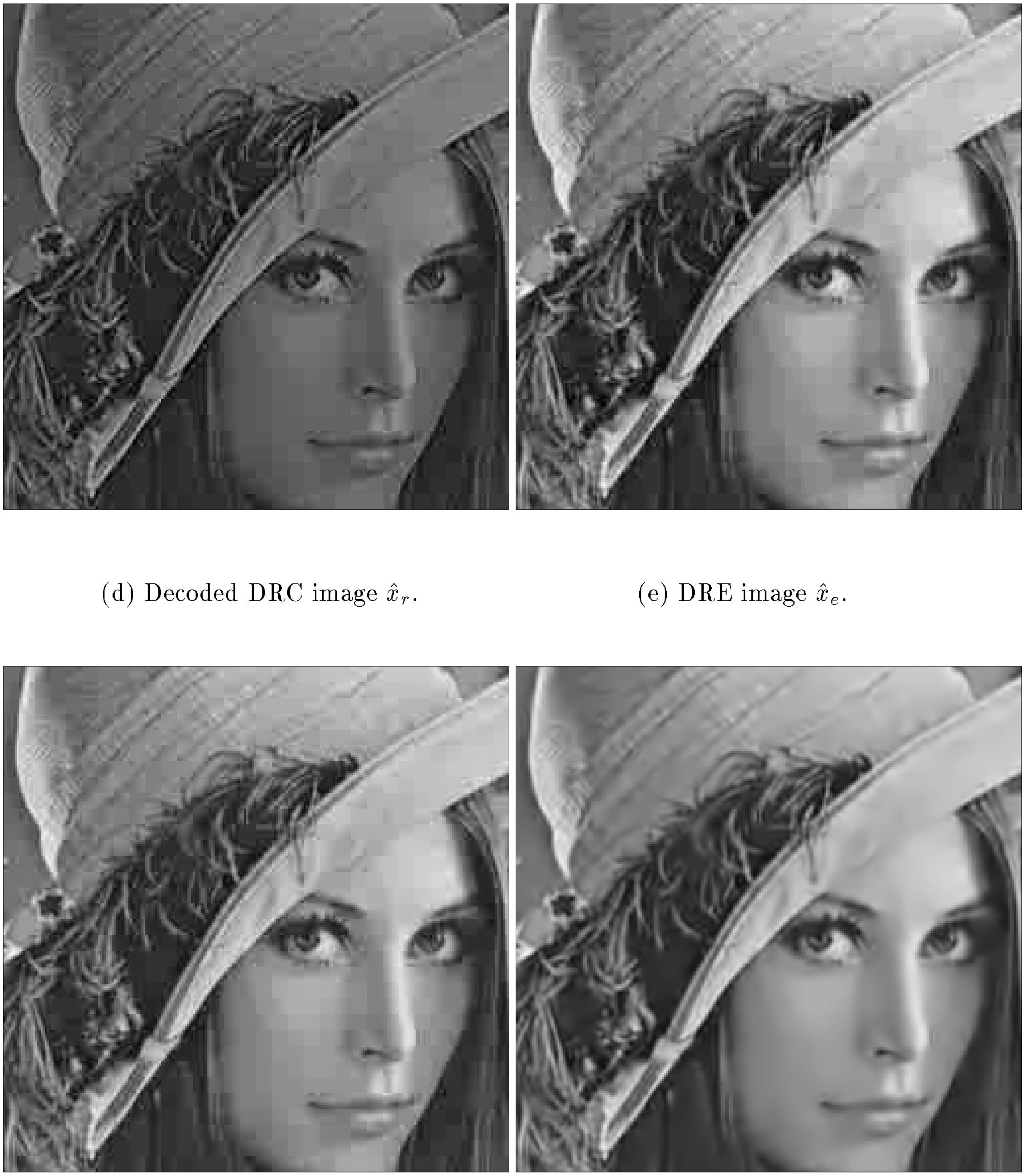


(a) Original image x.

(b) Preemphasized image x_e.

(c) DRC image x_r.

Figure 5.3: Segments of the 512×512, 8-BPP Lena image at the IPF encoder.



(d) Decoded DRC image $\hat{x}_r$.

(e) DRE image $\hat{x}_e$.

(f) Deemphasized image $\hat{x}$.

(g) JPEG-IPF-ESAP image $\tilde{x}$.

Figure 5.4: Segments of the 512×512, 0.25-BPP Lena image at the IPF decoder.

where the dB gain G applies to both dimensions, $P(\omega, G)$ is the preemphasis filter and $D(\omega, G)$ is the deemphasis filter. $P(\omega, G)$ is designed to obey the preemphasis characteristic

$$p[n] = \delta[n] + g\, h[n] \Longleftrightarrow P(\omega) = 1 + g\, H(\omega). \tag{5.4}$$

From 5.4 above, the dB gain is given by

$$G_{dB} = 10 \log_{10}(1 + g) \tag{5.5}$$

since $p[n]$ is a non-causal even-symmetric real sequence. The 2-D separable pre-post filter inverse pair is given by

$$\begin{aligned} P(\omega_1, \omega_2, G)\, D(\omega_1, \omega_2, G) &= \\ P_1(\omega_1, G)\, P_2(\omega_2, G)\, G_1(\omega_1, G)\, G_2(\omega_2, G) &= 1. \end{aligned} \tag{5.6}$$

Finally, companding obeys the following equations. For dynamic range *compression*

$$x_r = C\,(x_e - x_{e_{min}}), \tag{5.7}$$

and for dynamic range *expansion*

$$\hat{x}_e = \frac{1}{C}\hat{x}_r + x_{e_{min}}, \tag{5.8}$$

where

$$C = \frac{255}{x_{e_{max}} - x_{e_{min}}}. \tag{5.9}$$

At the *decoder*, the JPEG file $X_i(\mathbf{k})$ is decompressed into the $\hat{x}_r$ image shown in Figure 5.4(d), and then it is *dynamic range expanded* (DRE) into the $\hat{x}_e$ image of Figure 5.4(e). The image is next deemphasized with the filter $D(\omega, G)$ to create the $\hat{x}$ image shown in Figure 5.4(f). Finally it is ESAP postfiltered into the $\tilde{x}$ image to obtain further block smoothing as shown in Figure 5.4(g). The ESAP filter can be inserted in any of three possible positions: pos_1, pos_2 or pos_3. Depending on which position is selected, we could create the intermediate process images $\tilde{x}_r$, $\tilde{x}_e$ or $\tilde{x}$. Along with the JPEG coded image $X_i(\mathbf{k})$, we need to pass four overhead bytes to the decoder: the dynamic range compression parameters $(x_{e_{min}}, x_{e_{max}})$ and the pre-post filter parameters (ω, G). This overhead only amounts to about 0.0001 BPP for a 512×512 8-BPP gray-scale image at a 32:1 compression ratio (0.25 BPP). Please note that the encoder contains a decoder within itself.

To start the program, an initial guess of (ω_0, G_0) needs to be given. The program is invoked from the command line using:

```
> compl_model_enc
USAGE: compl_model_enc     x     H   W   R   wc    G  esap_pos N Gr gn_file
ex:    compl_model_enc lena.512 512 512 0.25 0.5 2.0     1    7 6  gn_N7_G6
```

A description of the input parameters follows:

`H` and `W` are the height and width of the input image `x`. `R` is the desired BPP rate, `(wc,G)` are initial condition preemphasis and deemphasis pre-post filter search parameters, `esap_pos` can only be {`1,2,3`}, `N` is the 2-D pre-post filters' order N×N, `Gr` is the pre-post filters' dB gain range $(-G_r..0..G_r)$, and finally, `gn_file` are the precomputed 2-D symmetric pre-post filters. They are precomputed with the external MATLAB function `compl_filt2file.m`. Only the IV quadrant of the filter coefficients is stored in the output file due to its four-quadrant symmetry.

It was found during testing of IPF that it can sometimes converge to *local* MSE minima, that is, the IPF MSE is not a true quadratic multidimensional function. This contrasts with the ESAP-only postfilter case shown in Figure 3.6, which involves truly monotonic 2-D functions. The IPF local minima behavior is shown in Figure 6.2 for SBC images, but it is similar for JPEG images. Equation 2.4 no longer holds for IPF. We attribute this non-monotonic behavior to the non-stationarity created by the changes in quantization tables for each iteration i with different values of (ω_i, G_i).

In order to speed up the search process, the user needs to use some insight into any previously obtained results to decide which new initial conditions could provide a smaller initial local MSE. In the case where there is no previous data, the algorithm can be initiated with (ω_c, G) = (0.01, 0.0). This initial search point corresponds to a pure JPEG operation, with neither preemphasis, nor dynamic range compression. It is also suggested to place ESAP in position 3. This generally provides the best postfiltering results since this allows the DRE and $D(\omega, G)$ functions to fully utilize the internal floating point precision available after decoding the $X_i(\mathbf{k})$ JPEG file.

5.4 Results

This section compares IPF with baseline JPEG, EZW and SPIHT algorithms. Tables 5.1 and 5.2 show the IPF results for the 512×512 8-BPP "Lena" image and Tables 5.3 and 5.4 show the results for the 512×512 "Barbara" image. All the images are *.pgm* formatted and are listed in Appendix A. The IPF images were obtained with the algorithm described in Figure 5.2. For comparison, the PSNR values obtained with baseline JPEG, Shapiro's Embedded ZeroTree Wavelet coder [41], and experimental results obtained with Said and Pearlman's Set Partitioning in Hierarchical Trees (SPIHT) subband coder [70] are shown in the third and the two rightmost columns, respectively. The EZW PSNR values are taken from the literature and their corresponding images are not available. The values (ω_{c_q}, G_q) are the *quantized* cutoff frequencies and gains for the pre-post filters, respectively. The expanded dynamic range is bounded by $(x_{e_{min}}, x_{e_{max}})$ which can normally be represented by two 12-bit signed integers. N is the 2-D pre-post filters' order $N \times N$, and finally, G_r is the pre-post filters' dB gain range $(-G_r..0..G_r)$.

5.5 Findings

This chapter is concluded with some observations based on the results shown in Tables 5.1 through 5.4 and observations made in Section 5.3.

1. The MSE(ω, G) of IPF is not a quadratic function of (ω, G). Therefore, trial-and-error or an exhaustive search must be used to obtain the absolute lowest

BPP	**IPF Parameters**	**Image/PSNR**			
	(ω_{c_q}, G_q) $(x_{e_{min}}, x_{e_{max}})$ N, G_r	JPEG	IPF	EZW	SPIHT
0.25	(0.0469, 8.9609) (-94, 1106) 13, 9.25	lena.25.jpg 31.68	lena.25.ipf 33.15	N/A 33.17	lena.25.sp 34.14
0.50	(0.0312, 7.8047) (-14, 976) 13, 9.25	lena.5.jpg 34.90	lena.5.ipf 36.34	N/A 36.28	lena.5.sp 37.25
1.00	(0.0469, 8.4375) (-20, 1019) 11, 10	lena1.0.jpg 37.96	lena1.0.ipf 39.28	N/A 39.55	lena1.0.sp 40.46

Table 5.1: JPEG, IPF, EZW and SPIHT comparison for 512×512 Lena image.

BPP	Δ_{IPF}	Δ_{EZW}	Δ_{SPIHT}
0.25	1.47	1.49	2.46
0.50	1.44	1.38	2.35
1.00	1.32	1.59	2.50

Table 5.2: IPF PSNR dB improvement for 512 × 512 Lena image.

BPP	**IPF Parameters**	**Image/PSNR**			
	(ω_{c_q}, G_q) $(x_{e_{min}}, x_{e_{max}})$ N, G_r	JPEG	IPF	EZW	SPIHT
0.25	(0.0469, 7.5938) (-195, 1021) 9, 9	barb.25.jpg 25.02	barb.25.ipf 27.23	N/A 26.77	barb.25.sp 27.40
0.50	(0.0312, 4.3125) (-33, 562) 7, 6	barb.5.jpg 28.27	barb.5.ipf 31.09	N/A 30.53	barb.5.sp 31.25
1.00	(0.0469, 3.5938 (-50, 451) 9, 5	barb1.0.jpg 33.10	barb1.0.ipf 35.87	N/A 35.14	barb1.0.sp 36.22

Table 5.3: JPEG, IPF, EZW and SPIHT comparison for 512×512 Barbara image.

BPP	Δ_{IPF}	Δ_{EZW}	Δ_{SPIHT}
0.25	2.21	1.75	2.38
0.50	2.82	2.26	2.98
1.00	2.77	2.04	3.12

Table 5.4: IPF PSNR dB improvement for 512×512 Barbara image.

MMSE (maximum PSNR). However the local minima that may be encountered are usually very close to the absolute MMSE. Thus, a suboptimal convergence should be acceptable given its proximity to the MMSE. This nonmonotonic behavior is illustrated in Figure 6.2 which is also applicable to JPEG-IPF.

2. ESAP position 3 offers the best PSNR postfiltering results since the DRE and $D(\omega, G)$ functions fully utilize the internal floating-point precision available immediately after JPEG decoding.

3. IPF results are comparable in PSNR and subjective quality to the EZW/SPIHT subband coders. The IPF PSNR is slightly lower than SPIHT, depending on the image and bit-rate.

4. The binary search for an optimal Q-table given a desired bit rate using the `adaptQw()` function is the most time consuming task within the IPF algorithm. The second most computationally intense task is ESAP. For example, a typical encoding/decoding iteration with $N = 7$ takes about 16 minutes on an unloaded SUN SPARC 2. Close to 50% of this time is dedicated to obtain the image-adaptive Q-table, 25% for ESAP and the remaining processing time is used by the companding and pre-post filtering operations.

CHAPTER 6

Application of IPF and ESAP to SBC — Implicit SBC Coefficient Generation

Iterative pre-post filtering (IPF) of SBC images attempts to improve the subjective and objective quality of Said & Pearlman's *Set Partitioning In Hierarchical Trees* (SPIHT) octave-band subband coder [69]. This pre-post filtering algorithm for SBC is a basically a modification to the iterative pre-post filtering of JPEG images presented in Chapter 5.

6.1 IPF and ESAP applied to SPIHT SBC images

In the previous chapter, we stated that the fundamental idea behind IPF is a tradeoff between the amplitude of the low frequency and the high frequency DCT coefficients to raise the low level, high frequency components of the signal above the blocking noise. However, since SBC does not introduce blocking, this algorithm does not improve the decoded image significantly for low bit rates near the 0.25 BPP range, as we will see shortly.

The SBC-IPF algorithm concepts are very similar to JPEG-IPF. High frequencies are preemphasized with respect to the low frequencies. Also, the overall dynamic range of the preemphasized image is compressed to an 8 BPP unsigned integer range with the intention to decrease the magnitude of the lower subbands while the magnitudes of the remainder higher frequency subbands are increased before encoding. The process is reversed at the decoder due to pre-post filters with the inverse identity relationship of Equation 5.1. In the JPEG case of Chapter 5, ESAP further adaptively smoothes out the blocking artifacts.

The SBC-IPF algorithm concepts are easily explained by referring to Figure 6.1. An 8-BPP gray-scale image x is preemphasized with a filter $P(\omega, G)$ to obtain a floating point image x_e, its dynamic range compressed (DRC) to an 8-bit unsigned integer range image x_r and then input to the SPIHT *codetree* subband coder.

The compressed stream $X(\mathbf{k})$ is decoded into the floating point image $\hat{x}_r$ using the *decdtree* program. Its dynamic range is expanded (DRE) into its original floating point range of $(x_{e_{min}}, x_{e_{max}})$ creating the floating point image $\hat{x}_e$. It is then deemphasized with the filter $D(\omega, G)$ to obtain $\hat{x}$. Given an initial search point (ω_0, G_0) on the MSE surface $\mu(\omega, G)$, the algorithm searches for the minimum mean square error MMSE$(x, \hat{x}, \omega, G)$.

The IPF-ESAP algorithm works well for the DCT but not so well for SBC. There are some issues related to adaptive postfiltering of SBC images. First, Figure 6.1 does not show an ESAP block after $D(\omega, G)$. The reason is that ESAP smoothed the reconstructed image $\hat{x}$ more than expected, therefore it was not retained as part of the algorithm.

In an experiment using a six-level octave band decomposition, the PSNR of the test image "lena" at 0.25 BPP decreased from 34.13 dB (SPIHT SBC-only value) to 33.63 dB (SPIHT-ESAP value) which is acceptable but not better than the original SBC image. To obtain the subband coefficients for ESAP postfiltering, we used the following four programs:

- analysis
- codetree_float
- decdtree_float
- synthesis.

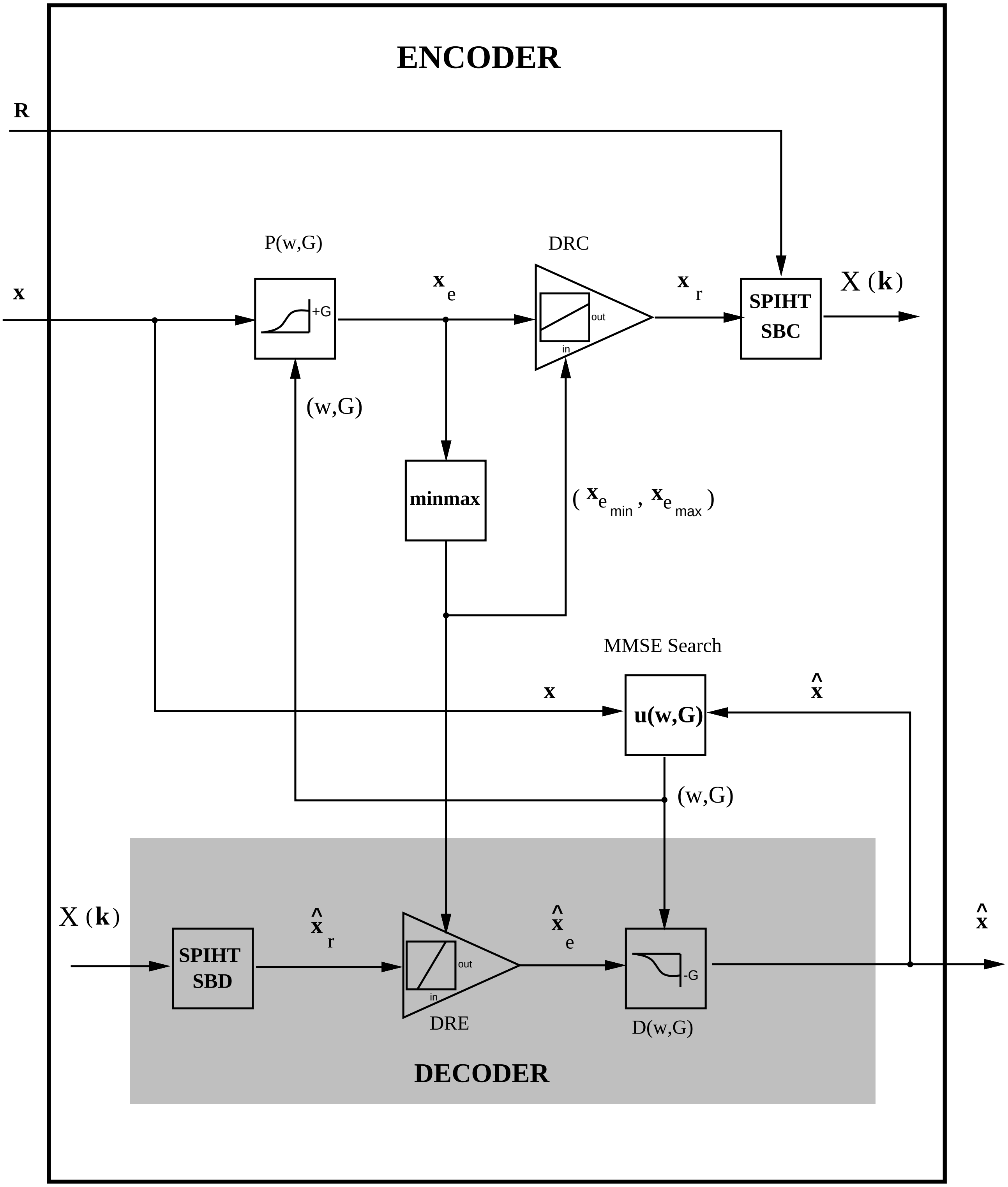


Figure 6.1: IPF of SPIHT SBC images.

To obtain the ESAP image we used a new revision of ESAP for SBC coefficients. The SBC-ESAP program files are also located in Appendix B. An example follows:

```
analysis lenna.512 lenna.6sub lenna.LLLLLL lenna.6sub.info 512 512 1 6
codetree_float lenna.6sub lenna.6sub.sph 512 512 1 0.25

decdtree_float -s lenna.6sub.sph lenna.6sub.rec 0.25
synthesis lenna.6sub.rec lenna.6sub.info lenna.rec.25.6sub

esap_r33 lenna.512 lenna.rec.25.6sub 512 512 lenna.6sub.rec
       lenna.rec.25.6sub.esap

psnr 512 512 lenna.512 lenna.rec.25.6sub.esap => 33.631176 dB
```

The second issue is the non-monotonicity of the SBC MSE surface $\mu(\omega, G)$, which as explained earlier, is also true for DCT images. The non-monotonicity is shown in Figure 6.2. Multiple local minima may prevent SBC IPF from converging to a true minimum MSE. The surface looks approximately smooth but it does have various local minima (see detail). During testing, the search sometimes stopped at one of these local dips.

In addition, we tried several other SBC-IPF-ESAP alternatives intended to improve the PSNR beyond the values obtained with the SPIHT SBC algorithm used alone. None of these were successful. Only the simple IPF SBC model without the ESAP block (Figure 6.1) was able to obtain a small improvement. The modifications included:

- Using 16 BPP preemphasized gray-scale input images to the SBC instead of 8 BPP images with no DRC/DRE

- Using passband pre-post filters $P(\omega, G)$ and $D(\omega, G)$ instead of the high pass filters currently used.

- Using non-inverse filters $P(\omega_1, G)$ and $D(\omega_2, G)$, where the cutoff frequency ω_1 is not equal to ω_2 $(\omega_1 \neq \omega_2)$. This is another form of passband preemphasis or deemphasis.

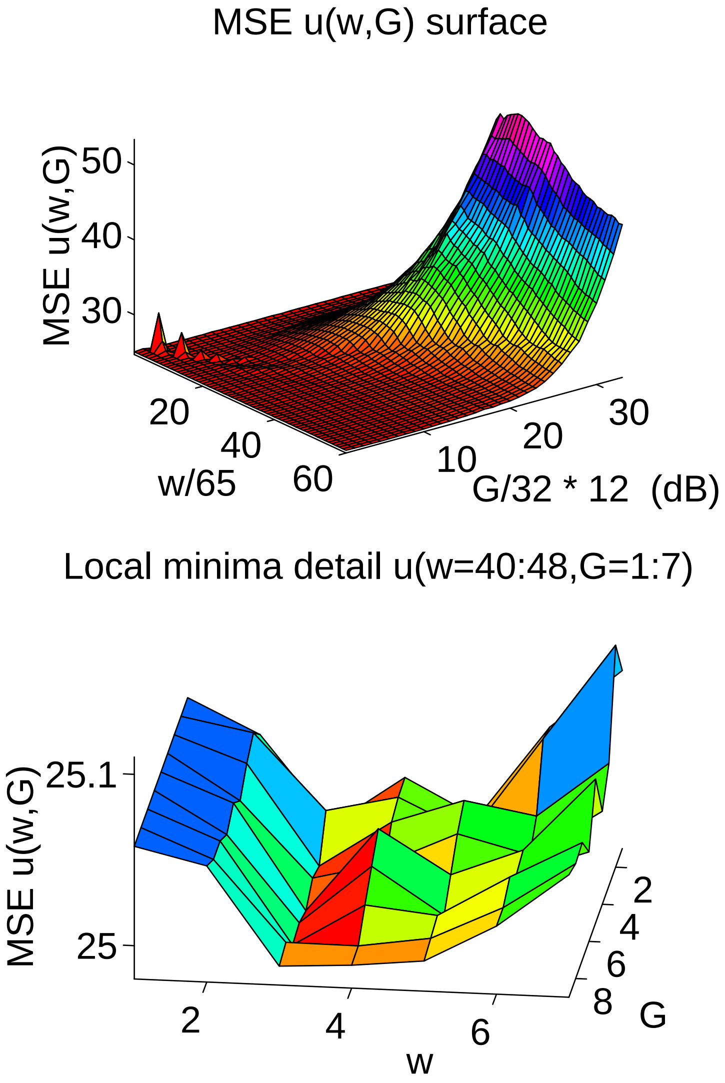


Figure 6.2: SBC MSE $\mu(\omega, G)$ surface for Lena @ 0.25 BPP.

- First deemphasizing, coding/decoding and then postemphasizing vs. current preemphasis — coding/decoding — deemphasis.

- Using a variety of pre-post filter sizes ranging from $M \times M = 3 \times 3, \ldots, 17 \times 17$. The best filters were in the 5×5 to 9×9 range.

One interesting space-frequency interpretation which did not provide any additional image enhancement is shown in Figure 6.3. From this figure, which contains the spatial frequency mapping for a 6-level octave band decomposition, we obtained the SBC coefficients to implicitly create the horizontal and vertical frequency images shown in Figures 6.4(a) and (b), respectively. These frequency images controlled a pixel-adaptive SBC version of the ESAP postfilter. Since this version did not improve the decoded image but did degrade it, this ESAP module was removed from Figure 6.1.

Figure 6.3 illustrates the subband structure. Each subband is pixel-replicated in the vertical and horizontal dimensions by an appropriate factor of {64, 32, ..., 2} to match the original image size of 512×512 pixels. The SBC-ESAP algorithm then looks at the *highest* magnitude coefficient for each pixel and assigns a SBC vertical or horizontal frequency bandwidth in the range $\{\frac{1}{64}, \frac{1}{32}, \ldots, \frac{1}{2}\}\pi$ to that pixel. From this information, the algorithm creates the frequency images shown in Figures 6.4(a) and (b). Next, the SBC-ESAP algorithm follows a procedure similar to the one explained for JPEG-ESAP in Section 3.1. The SBC coefficient-bandwidth relationships are shown in Table 6.1. The values in the table include vertical, horizontal and diagonal bands and need to be interpreted accordingly. For example, a pixel with a SBC bandwidth of (c_5, c_{10}) would be filtered with a separable pixel adaptive HPF with vertical frequency cutoff of $\frac{1}{16}\pi$ and a horizontal cutoff of $\frac{1}{4}\pi$. A pixel with a bandwidth of (c_9, c_9) will be filtered with a HPF with cutoffs of $(\frac{1}{8}\pi, \frac{1}{8}\pi)$.

Although it is unknown at this time of any other application that could benefit from the interpretation of Figure 6.4, it is shown for completeness. It shows the location of the high-frequency pixels in the image as derived directly from the SBC

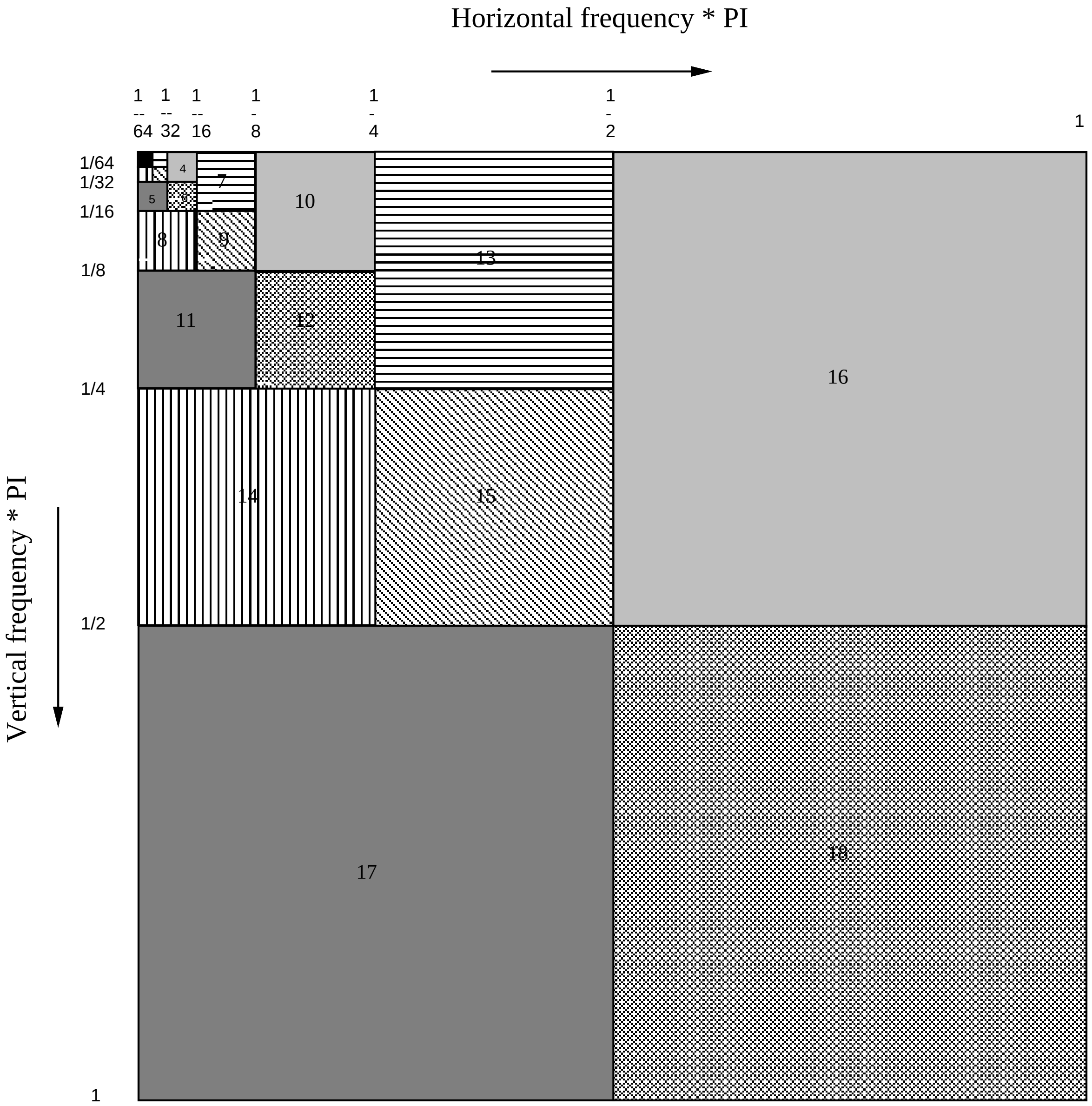
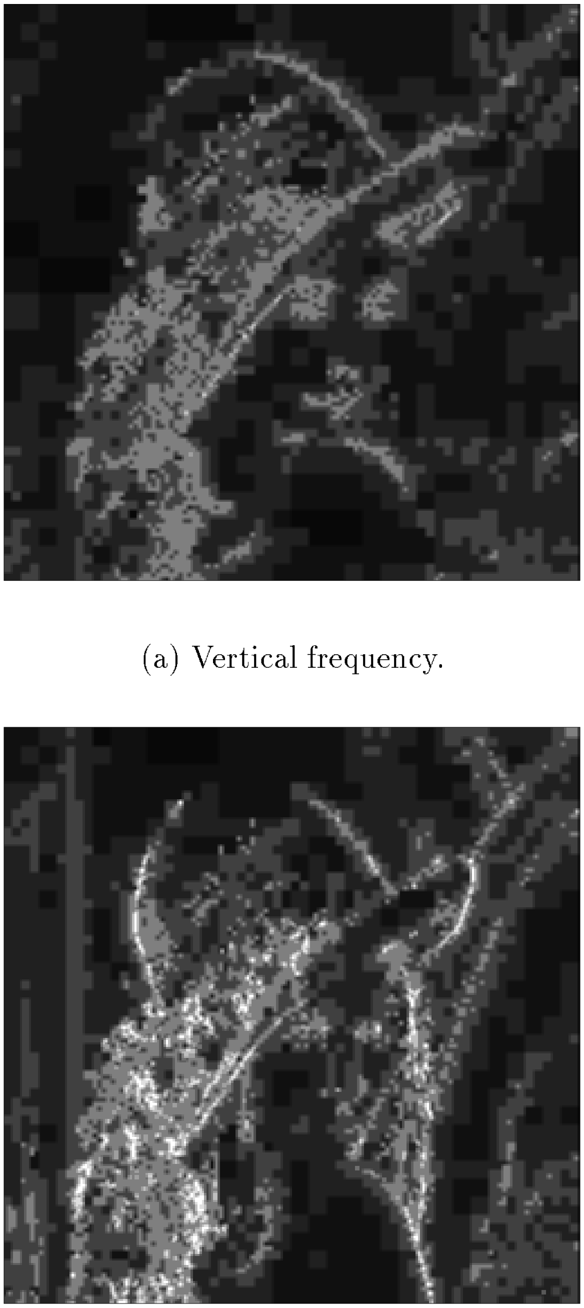


Figure 6.3: Spatial frequency mapping of a 6-level octave band SBC decomposition.



(a) Vertical frequency.

(b) Horizontal frequency.

Figure 6.4: Vertical and horizontal frequency images for a 6-level octave band SBC decomposition.

Highest SBC Coeff. Band Present	Normalized 1-D Bandwidth
c_0	0.015625π
c_1, c_2, c_3	0.03125π
c_4, c_5, c_6	0.0625π
c_7, c_8, c_9	0.125π
c_{10}, c_{11}, c_{12}	0.25π
c_{13}, c_{14}, c_{15}	0.5π
c_{16}, c_{17}, c_{18}	1.0π

Table 6.1: SBC coefficient-bandwidth relationships for a 6-level octave band decomposition.

coefficients. These images correlate well with the spatial edges and the highly textured areas, as expected. For further reference, the ESAP-SBC *.pgm* images and *.ps* diagram locations are listed in Appendix A. They are the following:

Image	**Description**
6-oct-dec.ps	Spatial frequency mapping of 6-level octave band decomposition
vertfreq.pgm	SBC vertical frequency image
horzfreq.pgm	SBC horizontal frequency image
lenna.25.6s.pgm	6-level oct. band SPIHT–34.13 dB
lenna.25.6s.esap.pgm	6-level oct. band SPIHT-ESAP–33.63 dB

6.2 Results

The actual results after applying the IPF-SBC algorithm in Figure 6.1 to the 512×512 8-BPP "Lena" and "Barbara" images are listed in Tables 6.2 and 6.3. The location of these *.pgm* images is also listed in Appendix A.

6.3 Findings

The application of IPF to SPIHT subband coded images did not improve their low bit rate quality. Indeed, it added too much external computational burden to the SPIHT algorithm. In its current form IPF-SBC has no practical use except for the

Rate	**SPIHT Image**	**IPF-SPIHT Image**	Δ_{IPF}
		$(\omega_c, G), N$	
	PSNR (dB)	PSNR (dB)	(dB)
0.25	lena.25.sp	lena.25.sp.ipf	
		(0.0176, 9.6875), 7	
	34.144	34.173	0.029
0.50	lena.5.sp	lena.5.sp.ipf	
		(0.0234, 10.0000), 7	
	37.251	37.262	0.011
1.00	lena1.0.sp	lena1.0.sp.ipf	
		(0.0225, 10.3125), 5	
	40.459	40.415	-0.044

Table 6.2: IPF-SPIHT results for the Lena image.

Rate	**SPIHT Image**	**IPF-SPIHT Image**	Δ_{IPF}
		$(\omega_c, G), N$	
	PSNR (dB)	PSNR (dB)	(dB)
0.25	barb.25.sp	barb.25.sp.ipf	
		(0.0156, 7.1250), 7	
	27.408	27.476	0.068
0.50	barb.5.sp	barb.5.sp.ipf	
		(0.0469, 7.5000), 7	
	31.251	31.255	0.004
1.00	barb1.0.sp	barb1.0.sp.ipf	
		(0.0156, 7.8750), 7	
	36.218	36.233	0.015

Table 6.3: IPF-SPIHT results for the Barbara image.

theoretical relationships derived from the space-frequency information contained in the SBC coefficients shown in Figures 6.4(a) and 6.4(b). Specifically we found that:

1. ESAP block smoothing works well for JPEG images, for which it was designed. It does not work that well for SBC images where there are no abrupt high-frequency, out-of-band blocking artifacts to filter out.

2. Pre-post filtering (IPF) works very well for JPEG images. It raises JPEG's performance close to most recent SBC algorithms with respect to visual and objective quality.

3. IPF improvement for SBC images is very small and it is not worth the extra computation.

4. There is a need to consider other classes of images besides natural images. SAR images would be good candidates for SPIHT processing due to their wide spectrum and the excellent performance of the SPIHT coder. Most likely, IPF will not improve SPIHT-coded images significantly. We base this assumption on the results shown in Table 4.8.

CHAPTER 7

Conclusion

The ESAP algorithm introduced in Chapter 3 combines tractable MSE metrics, a priori HVS spatial frequency response knowledge and frequency domain analysis to obtain significant objective and subjective enhancement of JPEG images that are coded at low bit rates. DCT blocking, which manifests itself as side lobes in the frequency domain, can be adaptively postfiltered to simultaneously improve the objective and the perceptual quality of the decoded images. ESAP reduces the DCT blocking without significantly blurring the true image edges with no increase in the nominal bit rate.

Each block's frequency bandwidth $\omega_i[\mathbf{m}]$ is estimated from the decoded DCT coefficients. These bandwidths are then interpolated to control a 2-D adaptive nonlinear postfilter $h_{\mathbf{m},\mathbf{z}}[\mathbf{n}]$ that changes to a directional filter, a nondirectional filter, or an impulse $\delta[\mathbf{n}]$, depending on the respective edge, non-edge or texture classification of the current pixel as determined from a Canny edge detection stage. Edges are filtered alongside and little or no filtering is performed across the edge in agreement with the HVS frequency masking properties, subject to FIR filter design constraints. The postfilter is completely characterized by the dequantized DCT coefficients and requires no side information, except for an optional image-dependent ESAP overhead if MMSE optimality is desired. This overhead is on the order of ≤ 0.0001 BPP for a 512×512 image coded at 0.25 BPP.

Most of the ESAP's PSNR gain is obtained from non-edge postfiltering. Edges and textures generally represent a small percentage of the pixels in an image coded at a low bit rate. Therefore, a simple nondirectional non-edge pixel postfilter that requires no overhead at all is amenable for fast software implementations, yet still provides a 1–1.5 dB PSNR gain. This algorithm is listed in Appendix B. On the other hand, more complex configurations that provide MMSE optimality at the expense of a small overhead are shown in Figures 3.1 and 5.2.

Any significant improvement ($\geq$ 3 dB PSNR) requires a combination of adaptive quantization or joint optimization along with pre-post filtering. Postfiltering or preprocessing used individually will only achieve 1–2 dB gain.

ESAP performance compared well with other image coding methods. We were able to improve upon JPEG, AQ, JO, VQ, LOT, and POCS-coded images. Finally, it was found that at moderate (1.0 BPP) or higher bit rates, the ESAP's PSNR and quality improvement became increasingly indistinguishable to the viewer, thus its added pre- and post-processing cost was not justifiable. The typical PSNR improvement is image and bit rate dependent and ranged between 0.5–3.2 dB over baseline JPEG. ESAP can potentially be incorporated into JPEG or MPEG to minimize their low bit rate blocking distortion. This in turn, could help to promote wider usage of these already popular standards.

Regarding SAR images, we were able to improve the *numerical* objective PSNR performance of 4 and 2 BPP images by using ESAP but the *subjective* quality seemed unaffected to the human eye. For rates of 1 BPP or less, ESAP slightly improved the PSNR but not the visual image quality. The apparent reason for this finding is that SAR images are inherently high frequency, therefore, any high compression will zero-out the high frequency coefficients, thus making impossible any image enhancement.

The IPF algorithm presented in Chapter 5 further enhances JPEG and ESAP with image-adaptive coefficient quantization and pre-post filtering. IPF was able to improve JPEG's performance by up to 2.8 dB. As with ESAP, we retained full compatibility with the JPEG standard and all the pre- and postprocessing was external to the image coding process. The main limitation of the IPF algorithm is the necessary additional computation needed to adapt the Q-table and to perform the pixel-adaptive ESAP postfiltering.

IPF is based on the *dbx* audio noise reduction system. The algorithm preemphasizes high-frequency areas and then compresses the dynamic range of the resulting image. This has the net effect of raising the low level high-frequency signals above the *blocking noise*, which is analogous to the audio *hiss* noise. This forces larger amplitude high-frequency DCT coefficients at the expense of lower amplitude DC and low-frequency AC coefficients. To maintain a constant bit rate, we use image-adaptive DCT coefficient quantization based on Lagrange multiplier unconstrained MSE minimization and binary search of the MMSE value λ that satisfies a desired bit rate b. The decoder expands the dynamic range to its original range and then an inverse filter deemphasizes the image. Finally, we apply ESAP to minimize any resulting blocking.

In Chapter 6, a similar ESAP bandwidth estimation algorithm based on the implicit coefficients of the SPIHT subband coder was not able to improve in any significant manner the decoded SBC image after pre-post filtering, companding and postfiltering. It was believed prior to experimentation with the SPIHT SBC, that SBC-IPF could possibly improve the subjective image quality as well as the PSNR based on the improvement over JPEG images and the synthesized SBC images of Section 4.5. Since the decoded SPIHT subband image does not have blocking artifacts, it is now obvious that the ESAP algorithm is not well-suited for SBC images.

Regarding VQ, although it has not gained too much acceptance due to its onerous codebook generation computational complexity, it is quite possible that ESAP in conjunction with IPF could improve upon the performance of VQ based on the results discussed in Section 4.3.

Most of the algorithms discussed in this thesis are asymmetric, and external to the encoding algorithm. Therefore the *decoder's* performance is as fast as the original decoding algorithm, plus the postprocessing. Since we embedded modularity in the algorithms, the additional computational load will depend on the number of modules selected. The selectable modules include adaptive quantization, linear pre-post filters (two modules), dynamic range companding (two modules), and nonlinear ESAP.

There are no iterative searches at the decoder, just straightforward decoding plus one or more of the following steps: dynamic range expansion, deemphasis and adaptive postfiltering. We previously mentioned in Section 5.2 that DRE and deemphasis are fast operations. If we limit adaptive postfiltering to the non-directional case only, ESAP achieves a reasonably good response on the order of a minute for a 512×512 gray-scale image. Therefore, in terms of the criteria presented in Table 7.1, the real-time performance of the *decoders* for each algorithm discussed falls in the "good" (3) or "fast" (4) range. The objective and subjective performance is the same for both encoder and decoder, since the encoders contain a decoder within themselves to predict the decoder's performance and to search for the optimal set of pre- and postprocessing parameters. The objective quality is measured by the PSNR. The basis for comparison is the JPEG's PSNR which is assigned an "average" value of 2. The subjective quality is based on the scored opinions of the author and several viewers. Images with a subjective quality comparable to JPEG are assigned an "average" value of 2.

Table 7.2 compares all the algorithms discussed in this thesis from the *encoder's* performance point of view based on the ratings of Table 7.1. We use baseline JPEG, EZW and SPIHT SBC encoders as our references. These C language algorithms were executed on Sun SPARC 2 and SPARC 10 workstations. The SPARC 10 is roughly twice as fast as the SPARC 2. All the images considered are 512×512, 8-BPP grayscale, except the SAR images which are 1008×1008, 8-BPP images.

It should be noted that although we were able to substantially improve the objective and subjective quality of the JPEG coder through AQ, JO, ESAP, and IPF algorithms, we were not able to outperform some current SBC algorithms like SPIHT. Nevertheless, we were able to match, and in some instances surpass EZW, an earlier SBC algorithm which was initially published in 1993.

Objective PSNR quality
1. low ($<$ JPEG)
2. average ($\approx$ JPEG)
3. good ($>$ JPEG)
4. high ($\geq$ EZW/SPIHT SBC)

Subjective visual quality
1. low ($<$ JPEG)
2. average ($\approx$ JPEG)
3. good ($>$ JPEG)
4. high ($\geq$ EZW/SPIHT SBC)

Real-time performance
0. unknown
1. very slow (hours)
2. slow ($\leq$ one hour)
3. good ($\approx$ few minutes)
4. fast (seconds)

Table 7.1: Performance ratings for the *encoder's* algorithms.

Algorithm	**Ref. Sect.**	**Objective Quality**	**Subjective Quality**	**Real-time Perf.**
JPEG	4.1	2	2	**4**
JPEG-CBS	4.1	1	2	**4**
JPEG-ESAP (non-iterarative)	4.1	3	3	3
JPEG-ESAP (iterarative)	4.1	3	3	2
JPEG-AQ	4.1	3	3	2
JPEG-JO	4.1	3	3	2
JPEG-AQ-ESAP	4.1	**4**	**4**	1
JPEG-JO-ESAP	4.1	**4**	**4**	1
EZW	4.1	**4**	**4**	0
LOT	4.2	2	2	3
LOT-ESAP	4.2	3	3	3
VQ	4.3	1	1	0
VQ-ESAP	4.3	1	1	2
POCS	4.4	2	2	3
POCS-ESAP	4.4	1	3	2
Image sequence	4.6	2	2	**4**
Image seq.-ESAP	4.6	3	3	3
Adaptive ROS	4.7	2	2	2
JPEG YC_bC_r	4.8	2	2	4
JPEG YC_bC_r-ESAP	4.8	3	3	2
SAR-JPEG	4.9	2	2	**4**
SAR-AQ	4.9	**4**	2	2
SAR-AQ-ESAP	4.9	**4**	2	1
Weighted MSE	4.9.1	1	1	2
JPEG-IPF-ESAP	5.4	**4**	**4**	1
SPIHT	6.2	**4**	**4**	**4**
SPIHT-IPF	6.2	**4**	**4**	1

Table 7.2: Performance comparisons for the *encoder's* algorithms.

7.1 Contributions

This thesis demonstrates that there are a number of *external* image processing techniques that can be applied to the DCT-based JPEG Standard to improve its objective and subjective performance close to the level of current state-of-the-art subband coders. All of this can be done while maintaining full JPEG syntax compliance. The most promising techniques to achieve these results are a combination of pre- and postprocessing techniques including AQ or JO in conjunction with ESAP, or the iterative pre-post filtering (IPF) technique used in conjunction with ESAP. The disadvantage of the external pre- and postprocessing is its added processing cost as compared to pure baseline JPEG. Nevertheless, these techniques are very promising for very low bit rate JPEG image enhancement with possible application to low bit rate videoconferencing and MPEG. They could potentially be implemented in firmware and further optimized to minimize the real-time processing latency. Specifically, the contributions and findings of this research include the following:

- ESAP provides an optional external blocking reduction module fully compliant with JPEG syntax.

- ESAP achieves a higher PSNR and better subjective blocking reduction than JPEG CBS AC prediction.

- ESAP provides a higher PSNR and better blocking reduction than POCS [42, 44, 45].

- Pre- and postprocessed AQ-ESAP and JO-ESAP are objectively and subjectively comparable to EZW and SPIHT SBC performance. They do not match SPIHT real-time performance in their current state.

- Based on the *dbx* audio noise reduction system, we developed and modeled IPF; a JPEG-compliant pre-post filtering and companding system.

- AQ-IPF-ESAP pre-post processing also improves the PSNR and subjective quality of JPEG images to levels close to EZW and SPIHT SBC.
- ESAP can enhance other block-based coded images as well:
  - VQ
  - Color JPEG images
  - MPEG
- All the algorithms presented require negligible overhead ($\leq$ 0.0002 BPP for rates $\geq$ 0.25 BPP.)
- The range of improvement of 0.5–3.2 dB PSNR with respect to JPEG is comparable or better than other nonlinear methods [55, 58, 61].
- All the above algorithms could be implemented in S/W, F/W or H/W. They can be potentially paralleled due to DCT block structure.
- We found that pre-post processing does not improve SPIHT SBC images.

Currently, the main disadvantage of the ESAP, AQ, and IPF algorithms is their computational complexity. ESAP's complexity is $O((MN)^2)$ additions, $O((NM)^2/4)$ multiplications per adaptive convolution for an $N \times N$ image and an $M \times M$ filter. However, a parallel decoder implementation could probably respond to real-time demands. Another ESAP limitation is the MMSE search complexity necessary to converge to an optimal PSNR point $p(\mathbf{z}^*)$. Nevertheless, in Section 3.5 we propose an optimally convergent $O((2D-1)N)$ search that does not require an exhaustive $O(N^D)$ search ($D = 4$). Also the IPF algorithm requires preemphasis filtering as well as deemphasis postfiltering passes. These could be implemented with FFTs, but they still increase the processing latency. Another minor inconvenience is the nonmonotonic convergence of the IPF algorithm. Due to the adaptive DCT coefficient quantization it is possible to converge to a local minimum. These local minima do not differ significantly from each other; therefore a suboptimal minimum provides an acceptable solution.

7.2 Future Research

These are some of the areas that were uncovered during this research. They offer the possibility of further performance improvement beyond what has been presented in this thesis. They include:

- 2-D pre-post equalization filtering. This goes beyond the simple highpass pre-emphasis characteristic of Equation 5.4. We foresee two major challenges with this technique:
  - the design of efficient pre-post equalization filters.
  - the large experimentation effort that would be required to develop the necessary rules-of-thumb for generic equalization band gain or attenuation.
- The possibility of performing *pixel*-adaptive pre-post filtering with ESAP enhancement versus the current *image*-adaptive pre-post filtering. This would be analogous to Dolby-B and Dolby-C noise reduction. A significant challenge for this effort is how to develop the space-varying pre-post filtering rules for proper tracking at the decoder. The advantage of this model is that it would not require the computationally intensive iterations of IPF at the encoder. It would not require any overhead either. Additionally, it would be a symmetric codec.
- Improve IPF with JO versus current AQ. In Chapter 5 we used AQ. Based on the excellent performance of the JO-ESAP algorithm (Tables 4.1–4.4), there is the potential to further improve IPF with JO.
- Compare IPF against perceptual optimization of JPEG images [27].
- SPIHT coding of DCT coefficients
  - combine SPIHT DCT coefficient coding with arithmetic entropy coding.
  - could possibly outperform JPEG's RLC-Huffman entropy coding in quality and real-time response.

- Explore ways to improve real-time performance of AQ, JO, ESAP, and IPF.

APPENDIX A

Processed Images

The images processed with the algorithms presented in this thesis are available at the following URL:

http://nmsp.gsfc.nasa.gov/~linares/images.html

The 8-BPP gray-scale image format is *.pgm.* For 24-BPP color images, the image format is *.pnm.* The figures are in postscript (*.ps*) format.

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.1	Lena	512 × 512	8.0	∞	> 4
	Lena JPEG		0.25	31.68	2
	Lena CBS		0.25	31.64	2
	Lena ESAP		0.25	32.76	3
	Lena AQ		0.25	31.88	3
	Lena AQ-ESAP		0.25	33.01	4
	Lena JO		0.25	32.34	3
	Lena JO-ESAP		0.25	33.07	4

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.1	Lena JPEG	512 × 512	0.5	34.90	2
	Lena CBS		0.5	31.87	2
	Lena ESAP		0.5	35.59	3
	Lena AQ		0.5	35.48	3
	Lena AQ-ESAP		0.5	36.23	4
	Lena JO		0.5	35.96	3
	Lena JO-ESAP		0.5	36.45	4
	Lena JPEG	512 × 512	1.0	37.96	2
	Lena CBS		1.0	37.95	2
	Lena ESAP		1.0	38.37	3
	Lena AQ		1.0	38.88	3
	Lena AQ-ESAP		1.0	39.23	4
	Lena JO		1.0	39.58	3
	Lena JO-ESAP		1.0	39.61	4

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.1	Barbara	512 × 512	8.0	∞	> 4
	Barbara JPEG		0.25	25.02	2
	Barbara CBS		0.25	25.01	2
	Barbara ESAP		0.25	25.79	3
	Barbara AQ		0.25	26.02	3
	Barbara AQ-ESAP		0.25	26.96	4
	Barbara JO		0.25	26.66	3
	Barbara JO-ESAP		0.25	27.05	4
	Barbara JPEG	512 × 512	0.5	28.27	2
	Barbara CBS		0.5	28.25	2
	Barbara ESAP		0.5	29.44	3
	Barbara AQ		0.5	29.99	3
	Barbara AQ-ESAP		0.5	31.03	4
	Barbara JO		0.5	30.63	4
	Barbara JO-ESAP		0.5	31.23	4
	Barbara JPEG	512 × 512	1.0	33.10	2
	Barbara CBS		1.0	33.09	2
	Barbara ESAP		1.0	34.01	3
	Barbara AQ		1.0	35.22	4
	Barbara AQ-ESAP		1.0	35.84	4
	Barbara JO		1.0	35.94	4
	Barbara JO-ESAP		1.0	36.33	4

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.2	Lena LOT	256 × 256	0.25	26.44	2
	Lena LOT-ESAP		0.25	27.38	3
	Lena LOT		0.5	29.18	2
	Lena LOT-ESAP		0.5	30.05	3
4.3	Lena VQ	512 × 512	0.25	29.28	1
	Lena VQ-ESAP		0.25	29.72	1
	Lena VQ		0.5	33.44	1
	Lena VQ-ESAP		0.5	33.91	1
4.4	Lena POCS	512 × 512	0.25	32.97	2
	Lena POCS-ESAP		0.25	29.66	3
4.5	Lena SBC	512 × 512	0.25	34.10	4
	Lena SBC-ESAP		0.25	34.24	4
	Lena SBC		0.5	36.89	4
	Lena SBC-ESAP		0.5	37.00	4
	Lena SBC		1.0	40.13	4
	Lena SBC-ESAP		1.0	40.13	4

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.6	Alexis.0 JPEG	256 × 256	0.67	38.91	2
	Alexis.1 JPEG			38.90	2
	Alexis.2 JPEG			38.96	2
	Alexis.3 JPEG			38.91	2
	Alexis.4 JPEG			38.94	2
	Alexis.5 JPEG			38.89	2
	Alexis.6 JPEG			38.94	2
	Alexis.7 JPEG			38.94	2
	Alexis.0 ESAP	256 × 256	0.67	39.17	3
	Alexis.1 ESAP			39.36	3
	Alexis.2 ESAP			39.25	3
	Alexis.3 ESAP			39.39	3
	Alexis.4 ESAP			39.30	3
	Alexis.5 ESAP			39.34	3
	Alexis.6 ESAP			39.23	3
	Alexis.7 ESAP			39.40	3
4.7	Lena JO-Adap-ROS	512 × 512	0.25	32.86	3
4.8.1	Lena YUV JPEG	512 × 512	0.21	30.91	2
	Lena YUV ESAP		0.21	31.30	3
	Lena YUV JPEG	512 × 512	0.30	31.77	2
	Lena YUV ESAP		0.30	32.44	3

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
4.9	hb06194 SAR	1008 × 1008	8.0	∞	> 4
	s4j SAR JPEG	1008 × 1008	3.98	38.31	2
	s4a SAR AQ		3.97	43.94	4
	s4e SAR ESAP		3.97	43.94	4
	s2j SAR JPEG	1008 × 1008	2.00	29.33	2
	s2a SAR AQ		2.00	31.87	4
	s2e SAR ESAP		2.00	31.87	4
	s1j SAR JPEG	1008 × 1008	0.99	26.48	2
	s1a SAR AQ		0.99	27.00	3
	s1e SAR ESAP		0.99	27.00	3
	s.5j SAR JPEG	1008 × 1008	0.49	24.84	2
	s.5a SAR AQ		0.49	24.91	1
	s.5e SAR ESAP		0.49	25.03	1
	s.25j SAR JPEG	1008 × 1008	0.24	23.59	2
	s.25a SAR AQ		0.23	23.61	1
	s.25e SAR ESAP		0.23	23.72	1

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
5.3	Preemph. x_e	512 × 512	12.0	–	–
	DRC x_r		12.0	–	–
	Dec. DRC $\hat{x}_r$		0.25	–	–
	DRE $\hat{x}_e$		12.0	–	–
	Deemph. $\hat{x}$		0.25	–	–
5.4	lena.25.ipf ($\tilde{x}$) IPF	512 × 512	0.25	33.15	4
	lena.5.ipf		0.5	36.34	4
	lena1.0.ipf		1.0	39.28	4
	barb.25.ipf	512 × 512	0.25	27.23	4
	barb.5.ipf		0.5	31.09	4
	barb1.0.ipf		1.0	35.87	4
5.4	lena.25.sp SPIHT	512 × 512	0.25	34.14	4
	lena.5.sp		0.5	37.25	4
	lena1.0.sp		1.0	40.46	4
	barb.25.sp SPIHT	512 × 512	0.25	27.40	4
	barb.5.sp		0.5	31.25	4
	barb1.0.sp		1.0	36.22	4
6.1	Figure 6.3 (*.ps*)	–	–	–	–
	Figure 6.4(a) (*.pgm*)	–	–	–	–
	Figure 6.4(b) (*.pgm*)	–	–	–	–
	lenna.25.6s.pgm SPIHT	512 × 512	0.25	34.13	4
	lenna.25.6s.esap.pgm (SPIHT-ESAP)	512 × 512	0.25	33.63	4

Sect.	**Image/Method**	**Dimensions**	**Rate**	**PSNR**	**Subj. Quality**
	Hyperlink		(BPP)	(dB)	(Tables 7.1 & 7.2)
6.2	Lena IPF-SPIHT	512 × 512	0.25	34.17	4
	Lena IPF-SPIHT		0.5	37.26	4
	Lena IPF-SPIHT		1.0	40.42	4
	Barbara IPF-SPIHT	512 × 512	0.25	24.48	4
	Barbara IPF-SPIHT		0.5	31.26	4
	Barbara IPF-SPIHT		1.0	36.23	4

APPENDIX B

Source Code

The C and MATLAB source code for the algorithms presented in this thesis are available at the following URL:

http://nmsp.gsfc.nasa.gov/~linares/source.html

These public-domain program files supplement the following principal topics of this thesis:

- JPEG DCT coefficient dump
- Adaptive Quantization of JPEG images
- ESAP gray-scale, color, and SAR image postfiltering
- DCT Iterative Pre-post Filtering
- SBC Iterative Pre-post Filtering

B.1 JPEG DCT Coefficient Dump

Reference: Section 4.1

```
The following IJG Version 5b JPEG files below were modified:

djpeg.c
jdcoefct.c -> jdcoefct.c.ESAP.color
jdcoefct.c.ESAP.color
jdcoefct.c.ORIGINAL -> jdcoefct.c
jdcoefct.c.ESAP.gray-scale (only for gray-scale images).
jmorecfg.h
```

The remaining IJG Version 5b JPEG files below were not modified:

```
Makefile
README
ansi2knr.1
ansi2knr.c
cderror.h
cdjpeg.h
change.log
cjpeg.c
ckconfig.c
coderules.doc
config.log
config.status*
configure*
example.c
filelist.doc
install.doc
jcapi.c
jccoefct.c
jccolor.c
jcdctmgr.c
jchuff.c
jcmainct.c
jcmarker.c
jcmaster.c
jcomapi.c
jconfig.bcc
jconfig.cfg
jconfig.dj
jconfig.doc
jconfig.h
jconfig.manx
jconfig.mc6
jconfig.sas
jconfig.st
jconfig.vms
jcparam.c
jcprepct.c
jcsample.c
jdapi.c
jdatadst.c
jdatasrc.c
jdcolor.c
jdct.h
jfdctfst.c
jfdctint.c
jidctflt.c
jidctfst.c
jidctint.c
jidctred.c
jinclude.h
jmemansi.c
jmemdos.c
jmemdosa.asm
jmemmgr.c
jmemname.c
jmemnobs.c
jmemsys.h
jpegint.h
jpeglib.h
jquant1.c
jquant2.c
jutils.c
jversion.h
libjpeg.a
libjpeg.doc
makcjpeg.st
makdjpeg.st
makefile.ansi
makefile.bcc
makefile.cfg
makefile.dj
makefile.manx
makefile.mc6
makefile.mms
makefile.sas
makefile.unix
makefile.vms
makljpeg.st
makvms.opt
rdbmp.c
rdcolmap.c
rdgif.c
rdjpgcom.c
rdppm.c
rdrle.c
rdtarga.c
```

jddctmgr.c
jdhuff.c
jdmainct.c
jdmarker.c
jdmaster.c
jdmerge.c
jdpostct.c
jdsample.c
jerror.c
jerror.h
jfdctflt.c
wrgif.c
wrppm.c

structure.doc
testimg.gif
testimg.jpg
testimg.ppm
testorig.jpg
testout.gif
testout.jpg
testout.ppm
usage.doc
wrbmp.c
wrtarga.c
wrjpgcom.c
wrrle.c

B.2 Adaptive Quantization of JPEG Images

Reference: Section 4.1

adaptQw_nr_malloc.c
qadaptw.c
nrutil.c

dct_table.h
mydecs.h
nrutil.h

Q.1
Q.255

B.3 ESAP Gray-scale Image Postfiltering

Reference: Section 4.1

esap_r32nr.c
aux_functions_esap_r32nr.c
nrutil.c

esap_r32.h
nrutil.h (nrutil.c and nrutil.h modified by I. Linares on 10-9-97)

B.4 ESAP YUV-3 Color Image Postfiltering

Reference: Section 4.8

```
PROCEDURE_color_esap

16x16_MCU_pad.c
cut_image.c
jdcoefct.ESAP.color.c
esap_r32nr.c
aux_functions_esap_r32nr.c
nrutil.c

esap_r32.h
nrutil.h
```

B.5 SAR-ESAP Image Postfiltering

Reference: Section 4.9

```
16to8bpp.c
esap_r32nr.c
aux_functions_esap_r32nr.c
nrutil.c

esap_r32.h
nrutil.h
```

B.6 JPEG Iterative Pre-post Filtering (JPEG-IPF)

Reference: Sections 5.2 and 5.3

```
adaptQw.c
qadaptw.c
dct_table.h
mydecs.h

esap_r32nr.c
```

```
aux_functions_esap_r32nr.c
nrutil.c
esap_r32nr.h
nrutil.h

compl_filt2file.m
compl_model.m

compl_model_enc.c
compl_model_enc_aux.c
compl_model_enc.h
```

B.7 SBC-ESAP Gray-scale Image Postfiltering

Reference: Section 6.1

```
esap_r33nr.c
aux_functions_esap_r33nr.c
nrutil.c

esap_r33.h
nrutil.h
```

B.8 SPIHT SBC Iterative Pre-post Filtering (SBC-IPF)

Reference: Section 6.1

```
EXAMPLE_codetree_float

float2int.c
compl_model_sbc.c
compl_model_sbc_aux.c
nrutil.c
compl_model_sbc.h
nrutil.h

compl_filt2file.m
```

```
NOTE: The SPHIT software,  Copyright (c) 1995, 1996 by Amir Said
      and William A. Pearlman,  can not be redistributed without
      the consent of the copyright holders.  For more information
      please refer to the following files:

        README
        SPIHT.doc
```

Bibliography

[1] W. B. Pennebaker, "Joint Photographic Expert Group (JPEG) Technical Specification," Draft Rev. 8, ISO/IEC JTC1/SC2/WG8 CCITT SGVIII, August 14, 1990

[2] G. K. Wallace, "The JPEG Still-Picture Compression Standard," *Communications of the ACM*, Vol. 34, No. 4, pp. 30–44, April 1991

[3] W. B. Pennebaker and J. L. Mitchell, "JPEG Still Image Data Compression Standard," ©1993 *Van Nostrand Reinhold*, NY

[4] T. G. Lane, "The Independent JPEG Group's Software," Releases 4a, 5b and 6 © 1990-95, Archive site: *ftp.uu.net*

[5] "The World Wide Web Consortium W^3C," *http://www.w3.org/pub/WWW*

[6] "Telecommunications Act of 1996," *http://www.technologylaw.com/techlaw/act.html*

[7] "ATSC Digital Television Standard," Document A/53, *http://atsc.org/standard.html*, Archive site: *ftp.atsc.org/pub/Standards/a_53.doc*

[8] S. K. Mitra and J. F. Kaiser, "Handbook for Digital Signal Processing," ©1993 *John Wiley & Sons, Inc.*

[9] B. Widrow and S. D Stearns. "Adaptive Signal Processing," ©1985 *Prentice-Hall, Inc.*, Englewood Cliffs, NJ

[10] J. M. Woram, "The Recording Studio Handbook," TK7881.4.W67, ©1979

[11] "Audio Engineering Handbook," ©1988 *McGraw-Hill Book Co. Inc.*, NY

[12] K. R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications," ©1990 *Academic Press, Inc.*, San Diego, CA

[13] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical Fast 1-D DCT Algorithms with 11 Multiplications," Proceedings IEEE *International Conference on Acoustics, Speech and Signal Processing*, pp. 988–991, Glasgow, Scotland, May 1989

[14] D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proceedings *IRE*, Vol. 40, pp. 1098–1101, 1962

[15] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Communications of the ACM*, Vol. 30, No. 6, pp. 520–540, June 1987

[16] G. G. Langdon, "An Introduction to Arithmetic Coding," *IBM Journal Research and Development*, Vol. 28, No. 2, pp. 135-149, March 1984

[17] J. L. Mannos and D. J. Sakrison, "The Effects of a Visual Fidelity Criterion on the encoding of Images," IEEE *Transactions on Information Theory*, Vol. rr-20, No. 4, pp. 525–536, July 1974

[18] D. J. Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission," IEEE *Transactions on Communications*, Vol. COM-25, No. 11, pp. 1251–1267, November 1977

[19] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," IEEE *Transactions on Communications*, Vol. COM-31, pp. 532–540, April 1983

[20] W. D. Hofmann and D. E. Troxel, "Making Progressive Transmission Adaptive," IEEE *Transactions on Communications*, Vol. COM-34, No. 8, pp. 806–813, August 1986

[21] B. Chitprasert and K. R. Rao, "Human Visual Weighted Progressive Image Transmission," IEEE *Transactions on Communications*, Vol. 38, No. 7, pp. 1040–1044, July 1990

[22] A. J. Ahumada Jr. and H. A. Paterson, "Luminance-Model-Based DCT Quantization for Color Image Compression," Proceedings *SPIE*, 1992

[23] J. A. Solomon, A. B. Watson, and A. Ahumada, "Visibility of DCT basis functions: Effects of contrast masking," Proceedings IEEE *Data Compression Conference*, pp. 361–370, Snowbird, UT, March 1994

[24] A. B. Watson, J. A. Solomon, and A. J. Ahumada, Jr., "Visibility of DCT basis functions: Effects of display resolution," Proceedings *Data Compression Conference*, pp. 371–379, Snowbird, UT, March 1994

[25] A. B. Watson, "DCT quantization matrices visually optimized for individual images," Proceedings SPIE *Human Vision, Visual Processing, and Digital Display IV*, 1993

[26] H. A. Peterson, A. J. Ahumada, and A. B. Watson, "An Improved Detection Model for DCT Coefficient Quantization," Proceedings SPIE *Human Vision, Visual Processing, and Digital Display IV*, 1993

[27] A. B. Watson and J. F. McGowan, "DCTune 1.0: Perceptual Optimization of JPEG Images," *http://vision.arc.nasa.gov/dctune/dctune1.0.html*

[28] S. J. P. Westen, R. L. Lagendijk, and J. Biemond, "Perceptual Optimization of Image Coding Algorithms," Proceedings IEEE *International Conference on Image Processing*, pp. 69–72, Vol. II, Washington, D. C., October 1995

[29] S. Wu and A. Gersho, "Rate-Constrained Picture-Adaptive Quantization for JPEG Baseline Coders," IEEE *International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, Vol. V, pp. 389–392, April 1993

[30] V. Ratnakar and M. Livny,"RD-OPT: An Efficient Algorithm For Optimizing DCT Quantization Tables," Proceedings IEEE *Data Compression Conference*, Snowbird, UT, March 1995

[31] M. Crouse and K. Ramchandran, "Joint Thresholding and Quantizer Selection for Decoder-Compatible Baseline JPEG," Proceedings IEEE *International Conference on Acoustics, Speech and Signal Processing*, pp. 2331–2334, Detroit, MI, May 1995

[32] M. Crouse and K. Ramchandran, "Joint Thresholding and Quantizer Selection for Transform Image Coding: Entropy-Constrained Analysis and Applications ro Baseline JPEG," IEEE *Transactions on Image Processing*, Vol. 6, No. 2, pp. 205–297, February 1997

[33] J. Canny, "A Computational Approach to Edge Detection," IEEE *Transactions on Pattern Analysis and Machine Intelligence*", Vol PAMI-8, pp. 679–698, November 1986

[34] S. Böe, "X-based Image processing Tools and Environment (XITE)," Archive site: *ftp.ifi.uio.no*, University of Oslo, Norway, February 1994

[35] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," ©1992 *Kluwer Academic Publishers*, Boston, MA

[36] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Image Coding Using Entropy-Constrained Residual Vector Quantization," IEEE *Transactions on Image Processing*, Vol. 4, No. 10, pp. 1349–1357, October 1995

[37] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Conditional Entropy-Constrained Residual VQ with Application to Image coding," IEEE *Transactions on Image Processing*, Vol. 5, No. 2, pp. 311–320, February 1996

[38] J. W. Woods and S. D. O'Neil, "Subband Coding of Images," IEEE *Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP 34, No. 5, pp. 1278–1288, October 1986

[39] M. J. T. Smith and T. P. Barnwell, III, "Exact Reconstruction Techniques for Tree-Structured Subband Coders," IEEE *Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 3, pp. 434–441, June 1986

[40] A. N. Akansu and M. J. T. Smith, "SUBBAND AND WAVELET TRANSFORMS — Design and Applications," ©1996 *Kluwer Academic Publishers*, Boston, MA

[41] J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," IEEE *Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445–3462, December 1993

[42] A. Zakhor, "Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding," IEEE *Transactions on Circuits and Systems for Video Technology*, Vol. 2, No. 1, pp. 91–95, March 1992

[43] S. J. Reeves and S. L. Eddins, "Comments on Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding," IEEE *Transactions on Circuits and Systems for Video Technology*, Vol. 3, No. 6, pp. 439–440, December 1993

[44] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Iterative Projection Algorithms for Removing the Blocking Artifacts of Block-DCT Compressed Images," IEEE *International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, Vol. V, pp. 405–408, April 1993

[45] K. Y. Kwak and R. A. Haddad, "Projection-Based Eigenvector Decomposition for Reduction of Blocking Artifacts of DCT Coded Image," IEEE *International Conference on Image Processing*, Vol. II, pp. 527–530, Washington, D. C., October 1995

[46] J. K. Su and R. M. Mersereau, "Post-Processing for Artifact Reduction in JPEG-Compressed Images," Proceedings IEEE *International Conference on Acoustics, Speech and Signal Processing*, Detroit MI, Vol. 4, pp. 2363–2366, May 1995

[47] "Coding of Moving Pictures and Associated Audio," *Draft of Standard MPEG 90/176 Rev. 1*—ISO-IEC JTC1/SC2/WG11, August 15, 1990

[48] D. Le Gall, "MPEG: A Video compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, No. 4, pp. 47–58, April 1991

[49] "Digital Processing of Video Signals—Video Coder/Decoder for Audiovisual Services at 56 to 1536 Kbit/s," *American National Standards Institute* (ANSI) Standard T1.p64-199x, August 15, 1990

[50] M. L. Liou, "Overview of the px64 Kbps Video Coding Standard," *Communications of the ACM*, Vol. 34, No. 4, pp. 60–63, April 1991

[51] "Encoding Parameters of Digital Television for Studios," *CCIR Recommendation 601* Document 11/1041-E, December 1985

[52] H. C. Reeve III and J. S. Lim, "Reduction of Blocking Effect in Image Coding," *International Conference on Acoustics, Speech and Signal Processing*, pp. 1212-1215, Boston MA, 1983

[53] B. Ramamurthi and A. Gersho, "Nonlinear Space-Variant Postprocessing of Block Coded Images," IEEE *Transactions on Acoustics, Speech and Signal Processing*, Vol.34, October 1986

[54] H. S. Malvar and D. H. Staelin, "The LOT: Transform Coding Without Blocking Effects," IEEE *Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 4, April 1989

[55] B. Zeng and A. N. Venetsanopoulos, "A JPEG-Based Interpolative Image Coding Scheme," *International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, Vol. V, pp. 393–396, April 1993

[56] R. L. Stevenson, "Reduction of Coding Artifacts in Transform Image Coding," *International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, Vol. V, pp. 401–404, April 1993

[57] Y. Wong, "Image Enhancement by Edge-Preserving Filtering," Proceedings IEEE *International Conference on Image Processing*, Vol II, pp. 522–524, November 1994

[58] W. E. Lynch, A. R. Reibman, and B. Liu, "Edge Compensated Transform Coding," Proceedings IEEE *International Conference on Image Processing*, Vol II, pp. 105–109, November 1994

[59] W. E. Lynch, A. R. Reibman, and B. Liu, "Post Processing Transform Coded Images Using Edges," Proceedings IEEE *International Conference on Acoustics, Speech and Signal Processing*, Detroit MI, Vol. 4, pp. 2323–2326, May 1995

[60] H. Greenspan and S.Akber, "Nonlinear Edge Enhancement," IEEE *International Conference on Image Processing*, Vol. I, pp. 179–182, Washington, D. C., October 1995

[61] A. Kundu, "Enhancement of JPEG Coded Images By Adaptive Spatial Filtering," IEEE *International Conference on Image Processing*, Vol. I, pp. 187–190, Washington, D. C., October 1995

[62] J. Jeong and B. Jeon, "Use of a Class of Two-dimensional Functions for Blocking Artifacts Reduction in Image Coding," IEEE *International Conference on Image Processing*, Vol. I, pp. 478–481, Washington, D. C., October 1995

[63] Y. Nakajima, H. Hori, and T. Kanoh, "A Pel Adaptive Reduction of Coding Artifacts for MPEG Video Signals," Proceedings IEEE *International Conference on Image Processing*, Vol II, pp. 928–932, November 1994

[64] I. Linares, R. M. Mersereau, and M. J. T. Smith, "Enhancement of Block Transform Coded Images Using Residual Spectra Adaptive Postfiltering," Proceedings IEEE *Data Compression Conference*, pp. 321–330, Snowbird, UT, March 1994

[65] I. Linares, "Optimal PSNR Estimated Spectrum Adaptive Postfilter for DCT Coded Images," Proceedings IEEE *International Conference on Acoustics, Speech and Signal Processing*, Detroit MI, Vol. 4, pp. 2387–2390, May 1995

[66] I. Linares, "JPEG Estimated Spectrum Adaptive Postfilter Using Image-Adaptive Q-tables and Canny Edge Detectors," Proceedings of the IEEE *International Symposium on Circuits and Systems*, Vol. 2, pp. 722–725, Atlanta, GA, May 12–15, 1996

[67] R. J. Clarke, "Spectral response of the discrete cosine and Walsh-Hadamard transforms," IEE *Proceedings*, Vol. 130, Pt. F, No. 4, June 1983

[68] P. E. Gill, W. M Murray and M. H. Wright, "Practical Optimization," ©1981 *Academic Press, Inc.*, San Diego, CA

[69] A. Said and W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, " IEEE *Transactions on Circuits and Systems for Video Technology*, Vol. 6, June 1996

[70] A. Said and W. A. Pearlman, "An Image Multiresolution Representation for Lossless and Lossy Compression," SPIE *Symposium on Visual Communications and Image Processing*, Cambridge, MA, November 1993

[71] Wright-Patterson Air Force Base Model Based Vision Lab, "MSTAR 16-BPP Public Clutter SAR Data,"
http://www.mbvlab.wpafb.af.mil/public/MBVDATA/mstrcd2.html

[72] Sandia National Laboratories, "How SAR Works,"
http://www.sandia.gov/RADAR/sar_sub/sar_intro1.html

[73] Alaska SAR Facility, "Frequently Asked Questions,"
http://www.asf.alaska.edu/user_serv/sar_faq.html

Vita

Irving Linares was born on June 9, 1954 in Cabo Rojo, Puerto Rico. He received his B.S.E.E. from the University of Puerto Rico at Mayaguez in 1977 and a M.S.E.E. from the Johns Hopkins University in Baltimore, MD in 1988. He received a Ph.D. degree from the Georgia Institute of Technology in 1998. His research interests include image and video coding, data compression and adaptive image filtering. He currently works as a computer engineer for the NASA Goddard Space Flight Center in Greenbelt, MD. His duties include support of the NASA's Network Control Center, the Tracking and Data Relay Satellite System and the Ground Network in the areas of computer systems, voice, video and data communications. He is recipient of the NASA Research and Study Fellowship and the NASA Silver Snoopy Award for his contributions to the U.S. manned space flight program. He is a member of Tau Beta Pi and the IEEE Signal Processing Society.

Adaptive Image Filtering for DCT Artifact Reduction

Irving Linares

120 pages
Directed by Dr. Russell M. Mersereau

This thesis investigates a number of image-adaptive, JPEG-compatible postfiltering and pre-post filtering methods designed to minimize the DCT *blocking* distortion. A pre-post filtering system uses inverse pair filters for high frequency preemphasis before encoding and high frequency deemphasis after decoding. The inverse pair pre-post filters are related by the inverse relationship $P(\omega) = 1/D(\omega)$, where P is the preemphasis filter and D is the deemphasis postfilter. A postfiltering system does not preemphasize the image before encoding. These techniques minimize the mean square error (MSE), improve the objective and subjective quality of low bit rate JPEG gray-scale images, and simultaneously enhance their perceptual visual quality. All the variants of the algorithms presented minimize the MSE below the level of baseline JPEG image compression, which is used as our comparison basis for similar bit rates. Convergence to a unique MMSE is possible for fixed quantization matrices, however, it cannot be guaranteed when image-adaptive quantization is jointly optimized under pre-post filtering.

We develop the theoretical basis of the *Estimated Spectrum Adaptive Postfilter* (ESAP) algorithm. ESAP is the main postfiltering algorithm used to minimize DCT blocking. ESAP utilizes either the default JPEG quantization table or image-adaptive DCT quantization matrices created in a preprocessing stage prior to image compression. At the decoder, the algorithm estimates 2-D pixel-adaptive bandwidths directly from the dequantized DCT coefficients to control a 2-D spatially-adaptive non-linear postfilter. Consistent with the human visual system tolerance to quantization errors in the high frequency regions, the algorithm performs directional filtering parallel to the edges and no filtering across the edges, subject to filter design constraints. Postfiltered images show minimal blurring of their true edges while blocking is significantly removed. ESAP relies on a DFT analysis of the DCT and is compliant with the coded stream syntax of the Independent JPEG Group (IJG) Version 5b Software.

Additionally, this thesis explores several other variants of the Estimated Spectrum Adaptive Postfilter applied to non-DCT coders such as vector quantization (VQ), subband coders (SBC), and Projection Onto Convex Sets (POCS). We compare the performance of these basic image coding methods against the same extended coders used with image preprocessing, ESAP postprocessing, or both, in a coder-compliant manner. These methods may or may not incorporate image-adaptive quantization and pre-post filtering. We also extend the concepts of the *dbx* audio noise reduction systems to model and demonstrate an *Iterative Pre-post Filter* (IPF). The IPF is applied to JPEG and to Set Partitioning In Hierarchical Trees (SPIHT) octave-band subband coders. We also process JPEG color images, synthetic aperture radar (SAR) images, and image sequences.

Typical PSNR improvement depends on the image, the encoding method, and the bit rate, and can range between 0.5–3.2 dB over baseline JPEG for 512×512 8-BPP gray-scale images. A comparison including all the treated techniques is presented at the conclusion of the thesis.